UNIVERSITY OF LEICESTER

Computational Task 3

For Data Mining and Neural Networks MA4022/ MA7022

Gleb Vorobchuk

Leicester

May 2018

UNIVERSITY OF LEICESTER

Task №1.

Data Evaluation.

For this task 4 time series was selected: Amazon,Ebay,Nvidia and Amd.

Data was selected in period from 2 January 2015 to 29 Decemberr 2017. Only dates and closing prices used.

Completeness Analysis

For all of the 4 data series was performed the same operations.

Since not every trading week starts from Monday and ends on Friday we need to change a concept.

Now we operate with open and close dates of the market.

OP is the day after closing day - Monday normally. CL is the day preceding Monday - Friday.

After that I found all the opening/closing dates in datasets.

Date	Close	Weekday	CL/OP		
02/01/2015	2.67	Friday	CL		
05/01/2015	2.66	Monday	OP		
06/01/2015	2.63	Tuesday			
07/01/2015	2.58	Wednesday			
08/01/2015	2.61	Thursday			
09/01/2015	2.63	Friday	CL		
12/01/2015	2.63	Monday	OP		
13/01/2015	2.66	Tuesday			
14/01/2015	2.63	Wednesday			
15/01/2015	2.52	Thursday			
16/01/2015	2.39	Friday	CL		
20/01/2015	2.24	Tuesday	OP	Monday	19/01/2015
21/01/2015	2.45	Wednesday			
22/01/2015	2.47	Thursday			
23/01/2015	2.45	Friday	CL		
26/01/2015	2.61	Monday	OP		

Then I found missing dates in pattern CL-OP, and filled in next or previous date.(Depends on CL or OP was missed).

Filtering all the missed dates I realised that most of them are similar for all of my datasets.

Then knowing that all the companies placed in US I supposed that missed dates are US National holidays.

Checking this assumption:

Monday	19/01/2015	Martin Luther King Jr. Day
Monday	16/02/2015	Washington's Birthday
Friday	03/04/2015	Good Friday
Monday	25/05/2015	Memorial Day
Friday	03/07/2015	Independence day
Monday	07/09/2015	Labour day
Friday	25/12/2015	Christmas

This assumption seems to be correct. All values that missed are US holidays.

Task №2.

Preprocessing.

Data preprocessing is one of the most crucial steps in data mining.

Data that we work sometimes could be inaccurate, incomplete or hard to work with because of the excessive dimensionality .

In order to fix this we can use several approaches:

1).Data cleaning -removing incorrect values. Adding missing values. Smooth the noise.

2).Integration - Collecting and wrapping information from different sources to provide users unified access to the information they need.

3).Transformation - Rescaling

4).Reduction - Reducing data records obtaining same results.[1]

In this task we need to deal with missing data so here are the few techniques that can can be used in Data Cleaning.

Partial imputation:

The expectation-maximization algorithm is an approach in which values of the statistics which would be computed if a complete dataset were available are estimated (imputed), taking into account the pattern of missing data. In this approach, values for individual missing data-items are not usually imputed.

Partial deletion:

Methods which involve reducing the data available to a dataset having no missing values include:

- Listwise deletion/casewise deletion
- Pairwise deletion

Full analysis:

Methods which take full account of all information available, without the distortion resulting from using imputed values as if they were actually observed:

- The expectation-maximization algorithm
- full information maximum likelihood estimation

Interpolation:

In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points.

In the comparison of two paired samples with missing data, a test statistic that uses all available data without the need for imputation is the partially overlapping samples t-test. This is valid under normality and assuming MCAR.[2]

In this task I filled the missing values of closing price by the previous one in time.

Here is the plot:

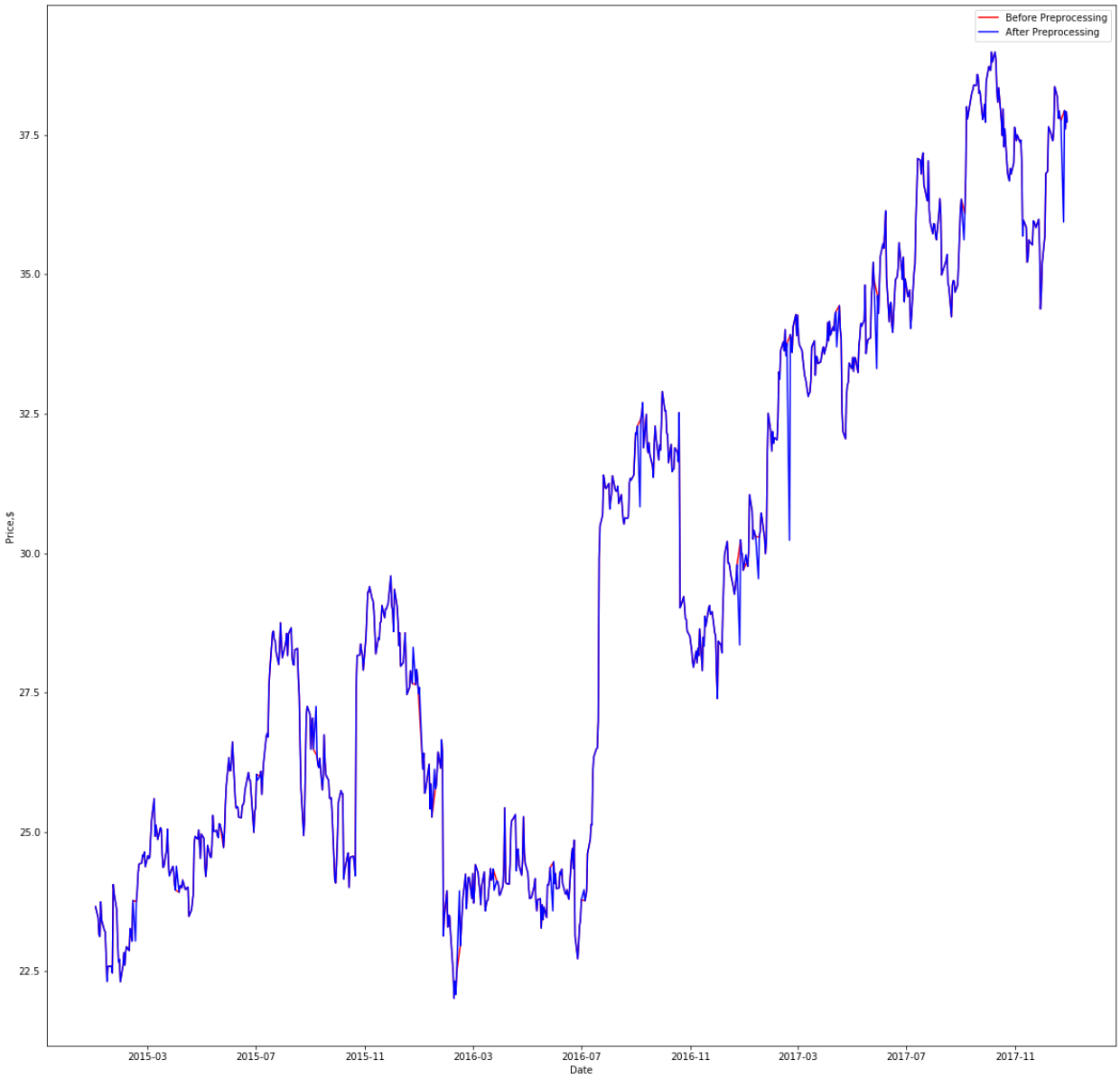


Figure 1. Ebay Time Series before and after filling.

Figure 2. Amazon Time Series before and after filling.

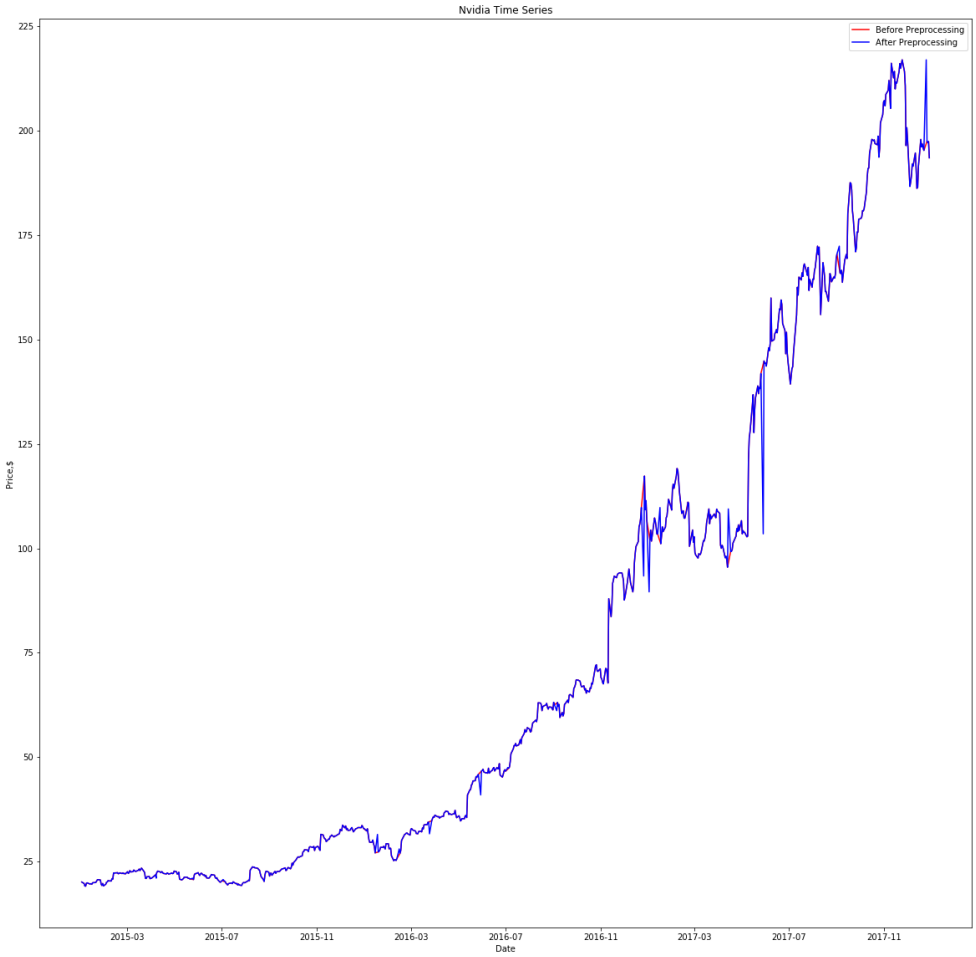


Figure 3. Nvidia Time Series before and after filling.

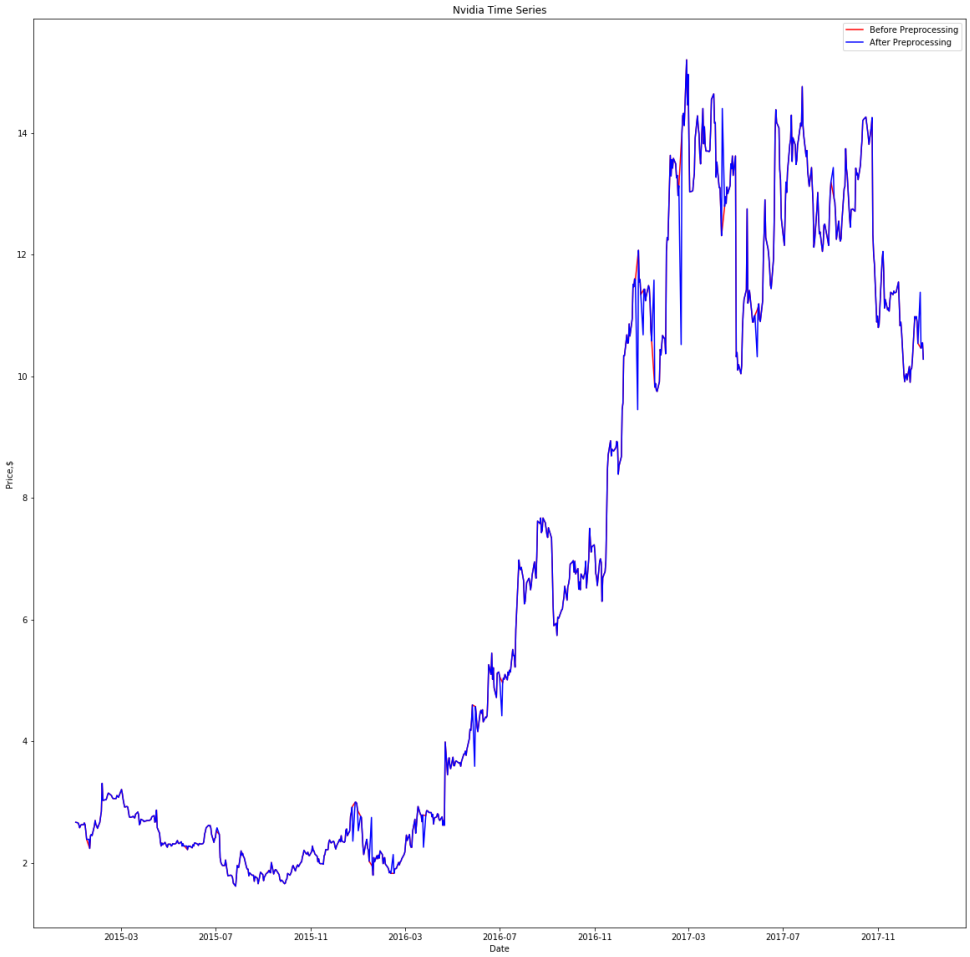


Figure 4. AMD Time Series before and after filling.

As we can see on figures 1-4 values are filled in.

After that I calculated mean and variance for each time series:

	Variance	Mean
Ebay	0.0002646036770954107	0.0006194221384244902
Amazon	0.00031397480204338826	0.0017672669409786795
Nvidia	0.000624380306323765	0.003001414220815962
Amd	0.0016981702366729212	0.0017879599305236865

Then I transformed time series to the dimensionless log-return using the rule:

$$y_i = \ln\left(\frac{x_i}{x_{i-1}}\right)$$

After that I normalised series to the z-score.

Here are the plots:

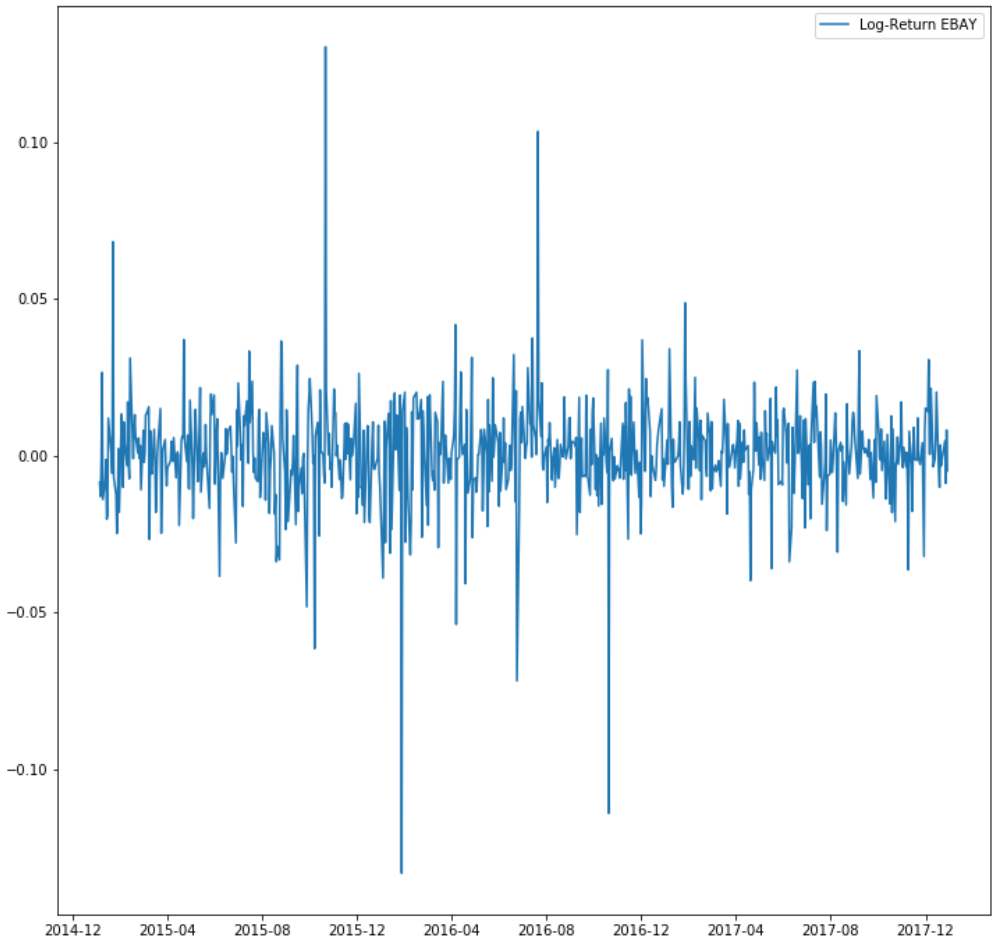


Figure 5.Log eBay

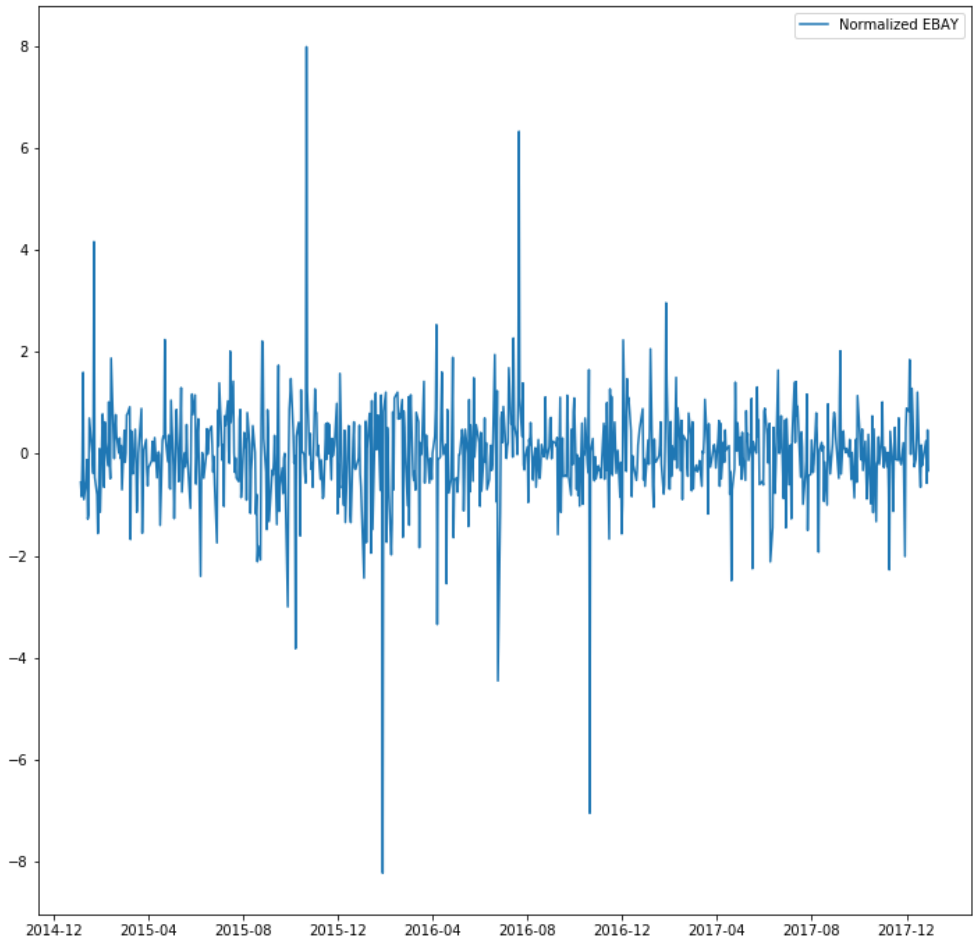


Figure 6. Normalised Ebay.

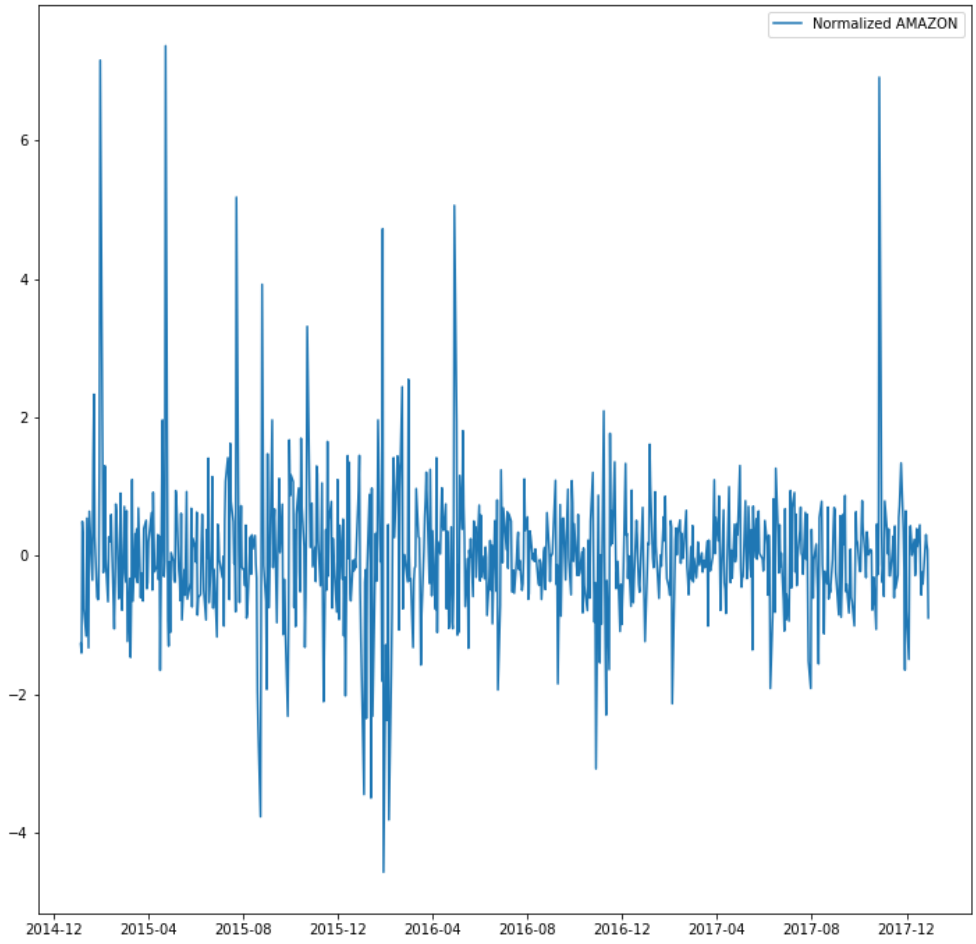


Figure 7. Log-Return Amazon.

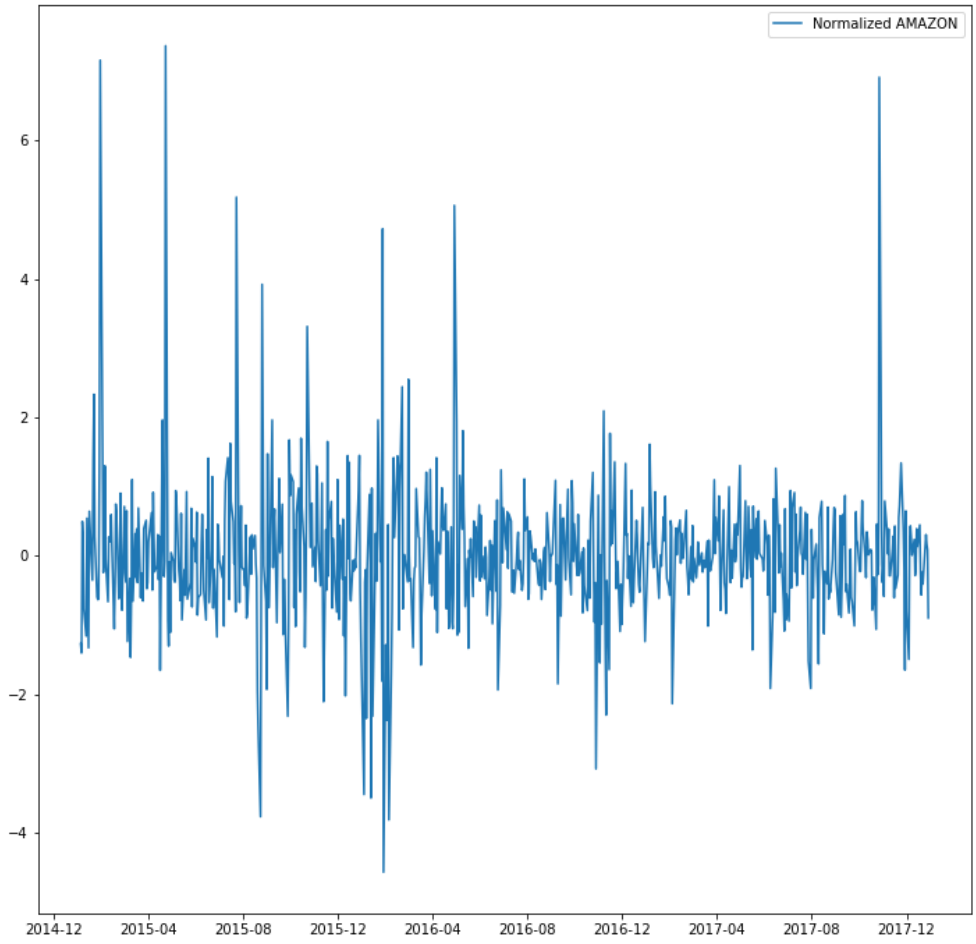


Figure 8. Normalised Amazon.

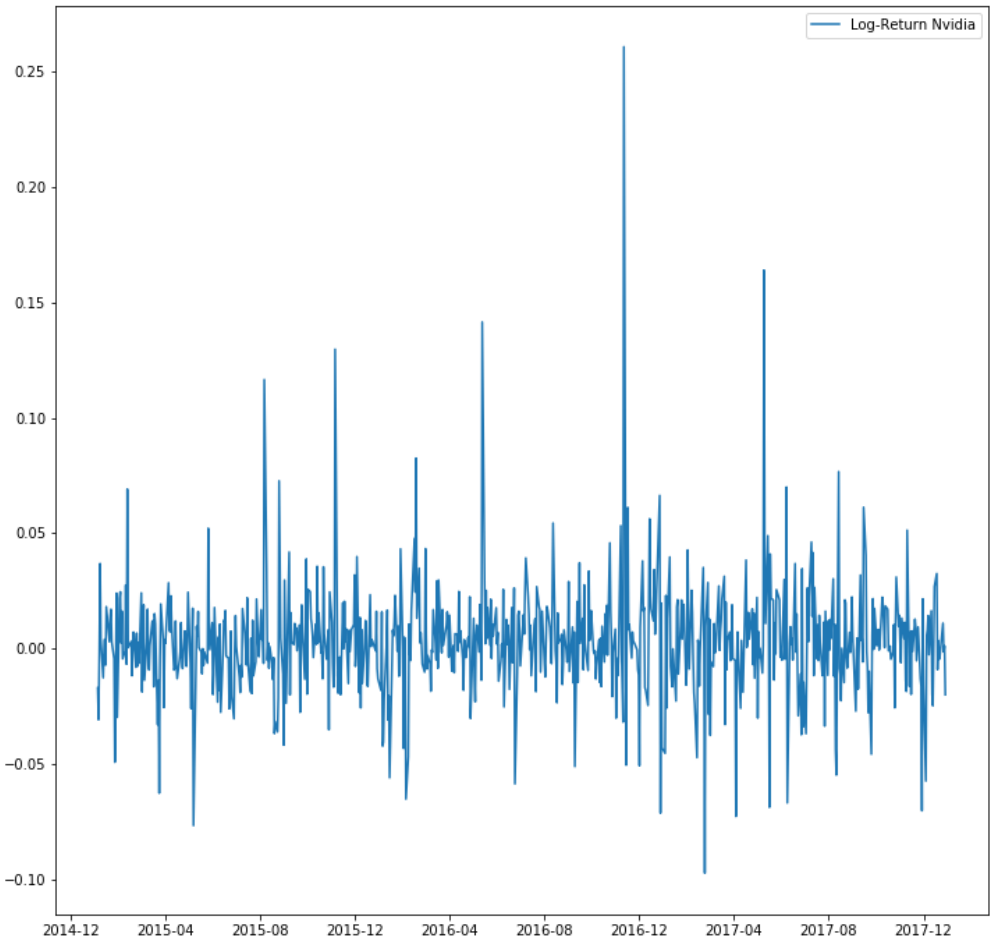


Figure 9. Log-Return Nvidia.

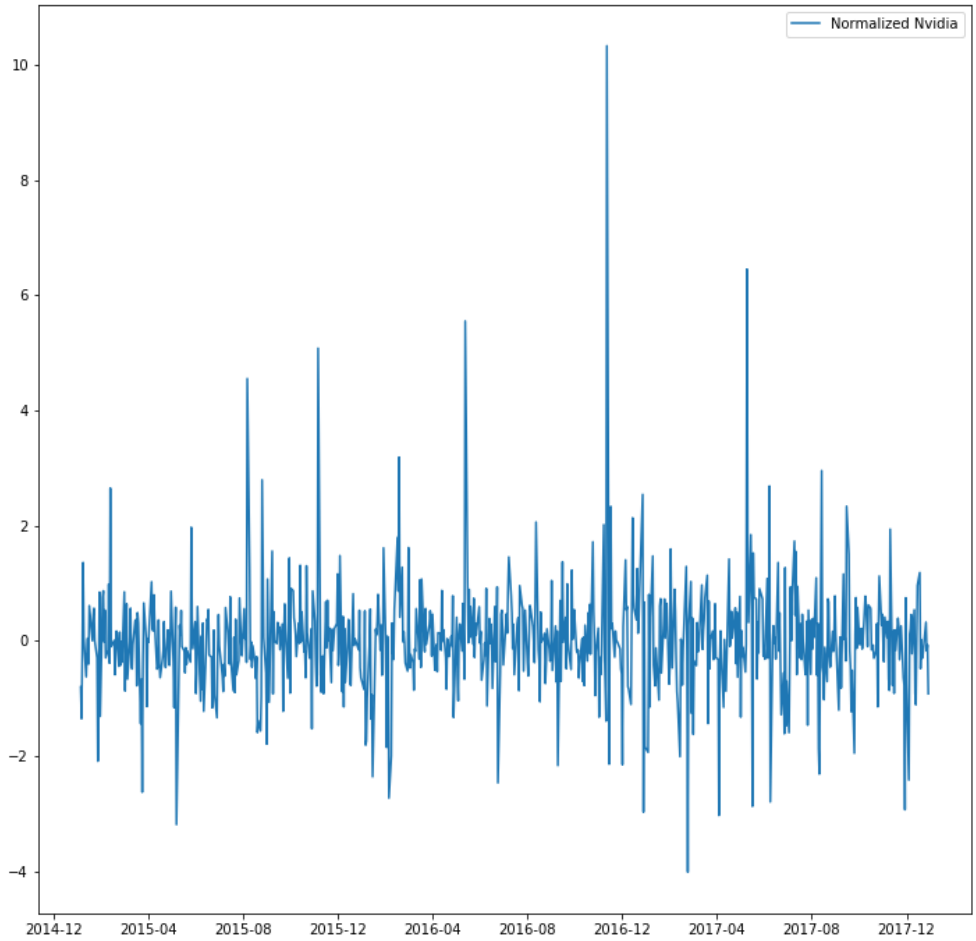


Figure 10. Normalised Nvidia.

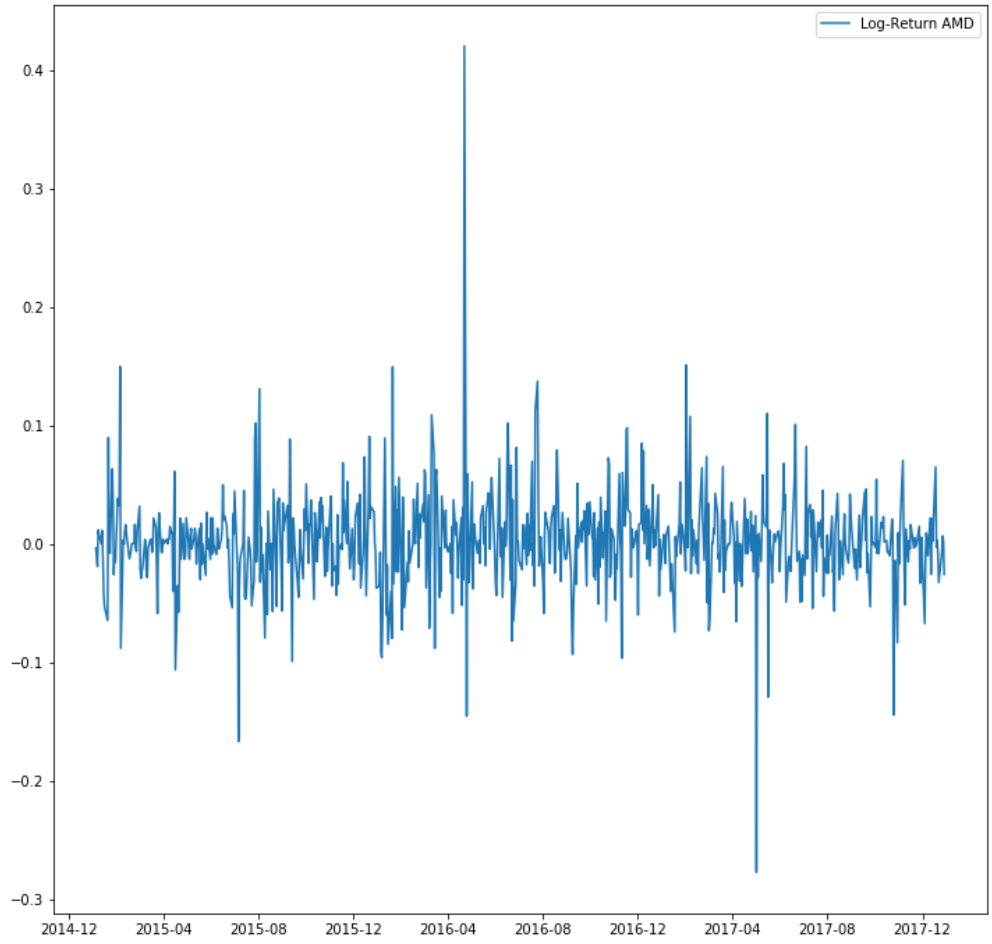


Figure 11. Log-Return AMD.

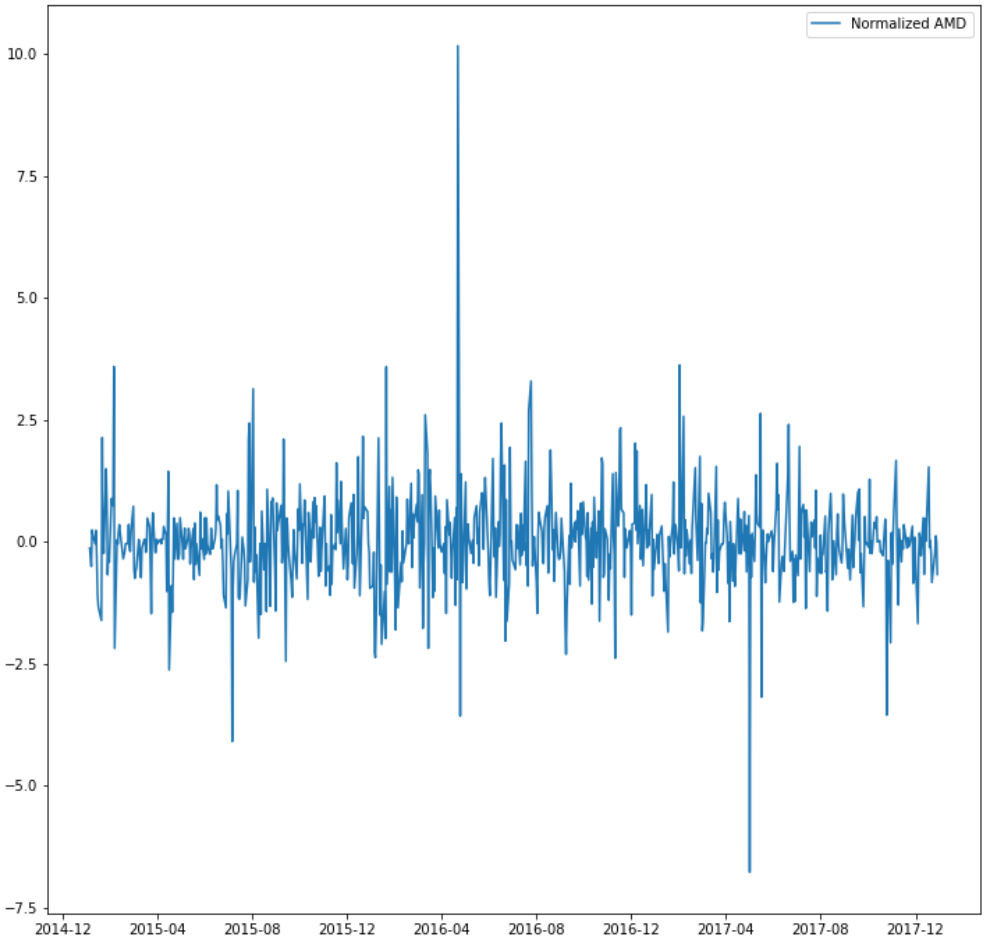


Figure 12. Normalised AMD.

UNIVERSITY OF LEICESTER

Task №3.

Segmentation.

First of all I've created a pairs of all possible combinations of points:

Normalized	Segments
-1.270526	[-1.270526,-1.404217]
-1.404217	[-1.404217,0.495647]
0.495647	[0.495647,0.284929]
0.284929	[0.284929,-0.767212]
-0.767212	[-0.767212,-1.159528]
-1.159528	[-1.159528,0.541863]
0.541863	[0.541863,-0.382164]
-0.382164	…..

Then I removed every second segment:

Segments	Segments
[-1.270526,-1.404217]	[-1.270526,-1.404217]
[-1.404217,0.495647]	
[0.495647,0.284929]	[0.495647,0.284929]
[0.284929,-0.767212]	
[-0.767212,-1.159528]	[-0.767212,-1.159528]
[-1.159528,0.541863]	
[0.541863,-0.382164]	[0.541863,-0.382164]
…..	

Here is the plot:

Then I performed linear regression for 4 data points of each pair of segments and calculated MSE.

Selecting pairs of segments with smallest MSE I'm merging the segments :

```
Min Index 347
Min Merge Cost: 0.000862829013202
Segments to merge
Segments    [0.793893063834, 0.400610513407]
Indices                          [695, 696]
Segments    [-0.0193962945298, -0.316194251368]
------
After merging
Segments    [0.793893063834, -0.316194251368]
Indices                          [695, 698]
```

Figure 13. Example of merging.

I performing this operations until minimal error is less than the threshold.

So for the different thresholds I have following results:

Treshold	Ebay Iterations	Amazon Iteration	Nvidia Iterations	AMD Iterations
0.04	52	54	46	61
0.1	107	115		120
0.25		208		200

Here are the plots:

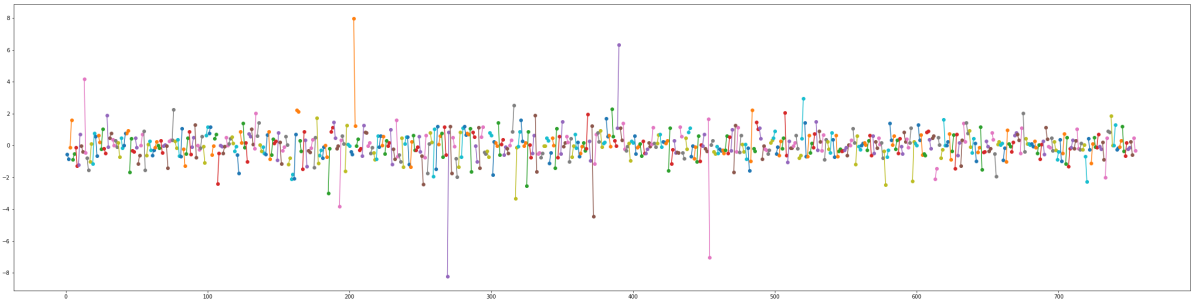


Figure 14. Ebay Segments Before Merging.

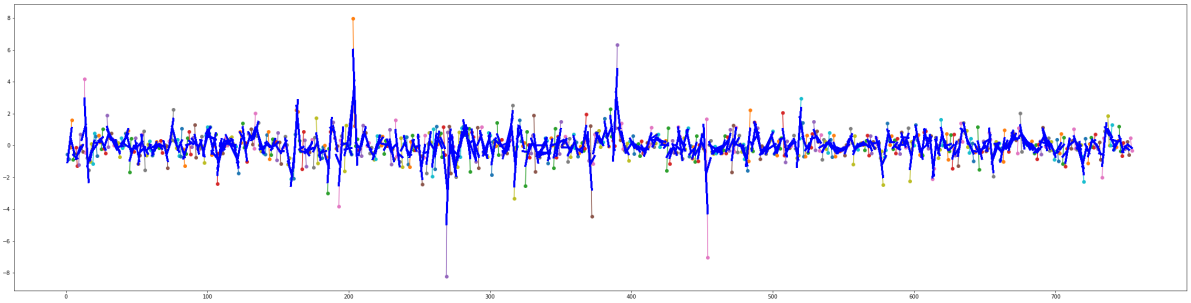


Figure 15. Ebay Segments with Regressions for Segments. with MSE threshold 0.04

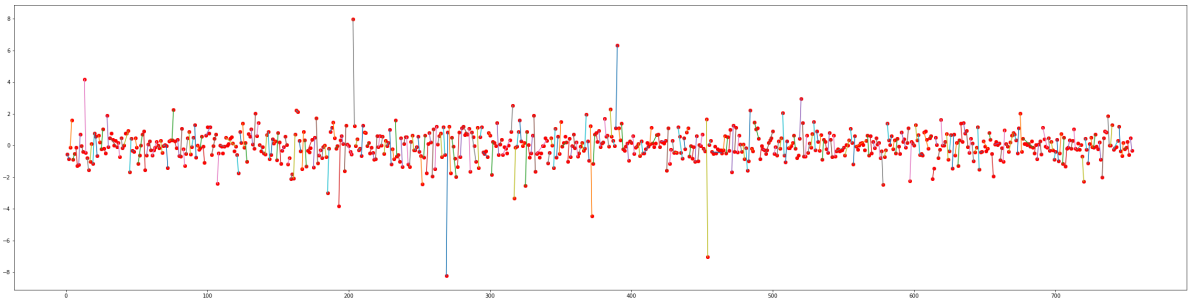


Figure 16. Ebay Segments after 52 Merging Iterations with MSE threshold 0.04.

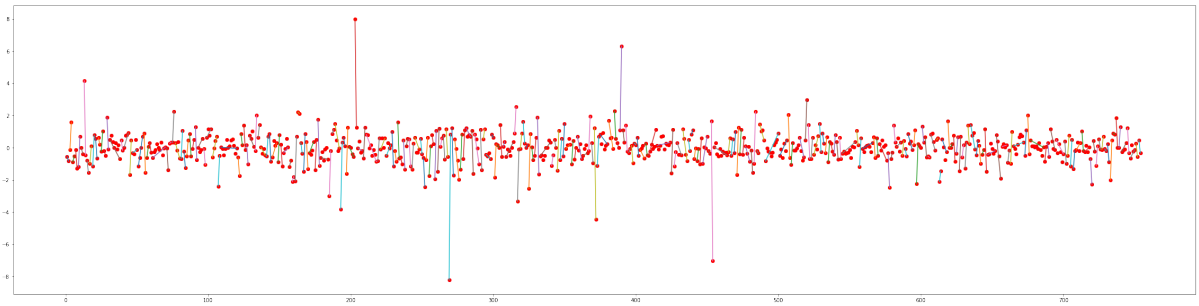


Figure 17. Ebay Segments after 107 Merging Iterations with MSE threshold 0.1.

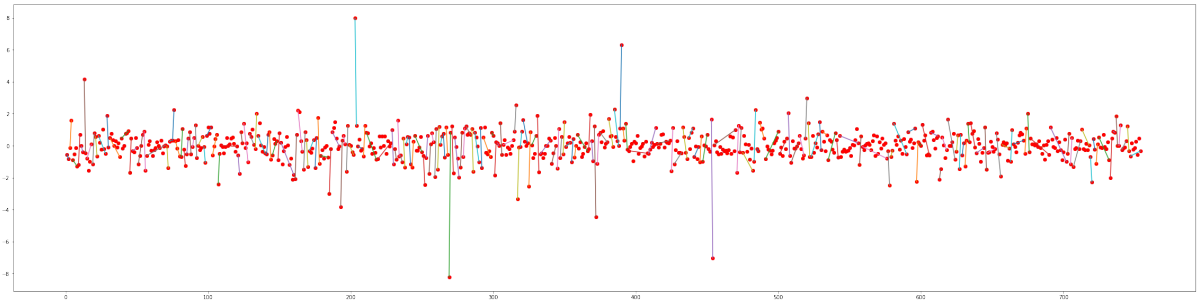


Figure 18. Ebay Segments after 54 Merging Iterations with MSE threshold 0.25.

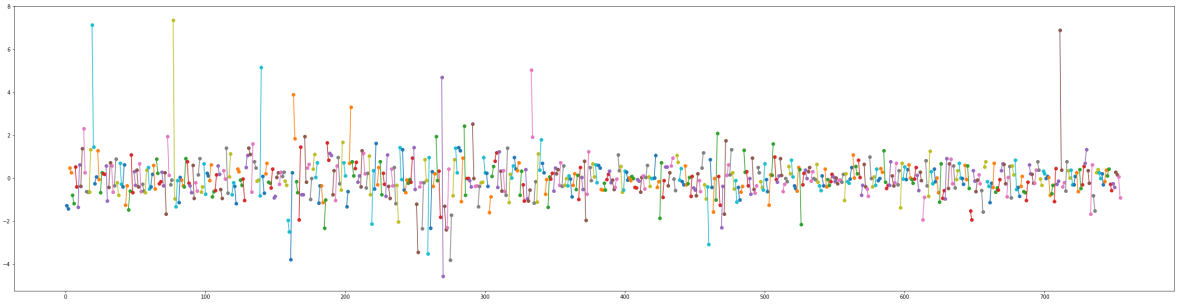


Figure 19. Amazon Segments Before Merging.

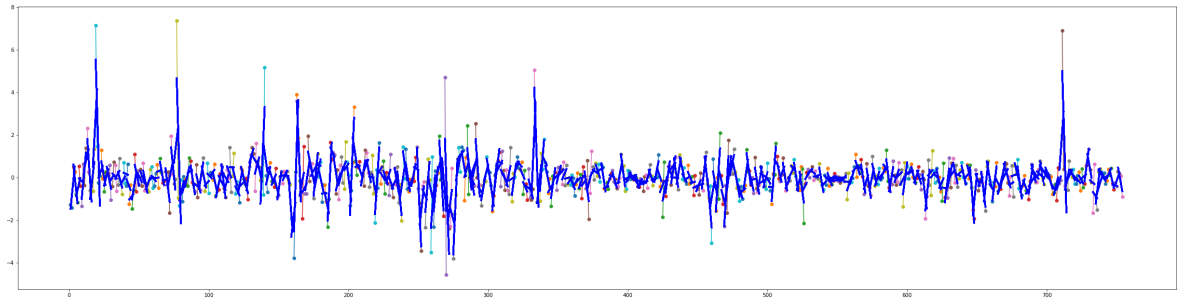


Figure 20. Amazon Segments with Regressions for Segments with MSE threshold 0.04.

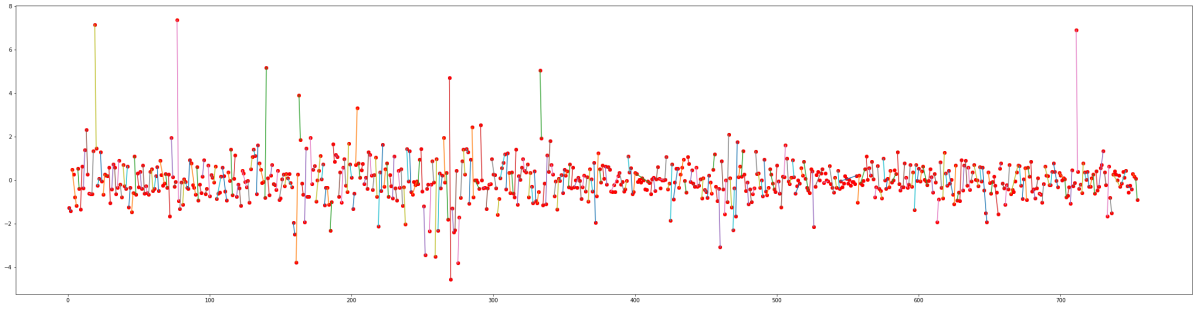


Figure 21. Amazon Segments after 54 Merging Iterations with MSE threshold 0.04.

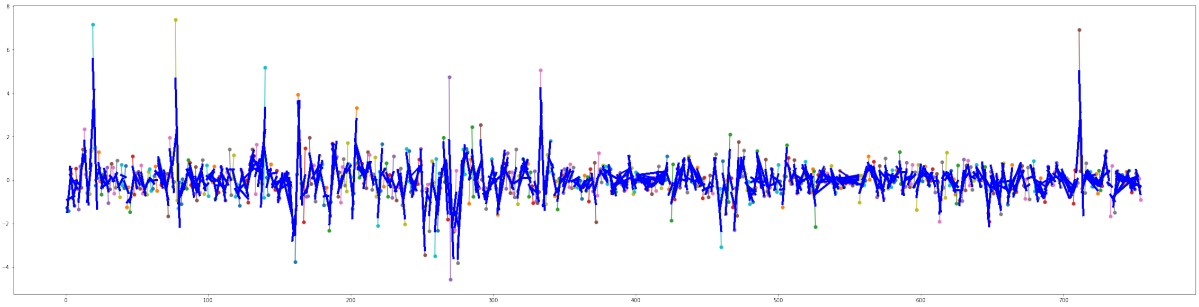


Figure 20. Amazon Segments with Regressions for Segments with MSE threshold 0.1.

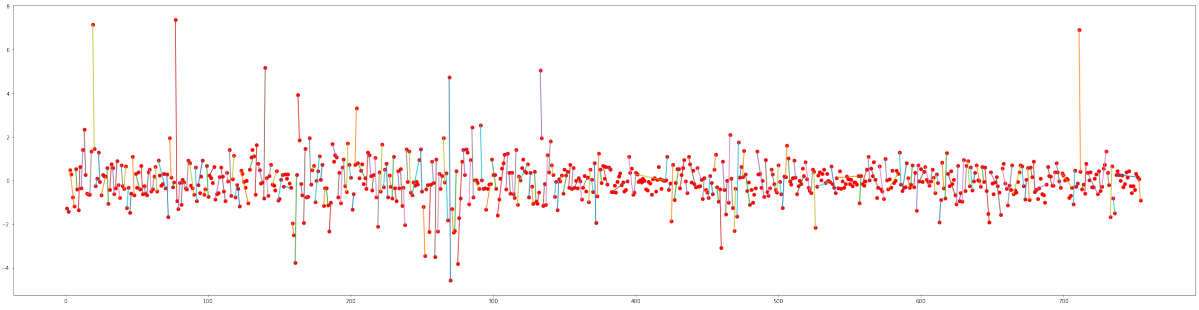


Figure 21. Amazon Segments after 115 Merging Iterations with MSE threshold 0.1.

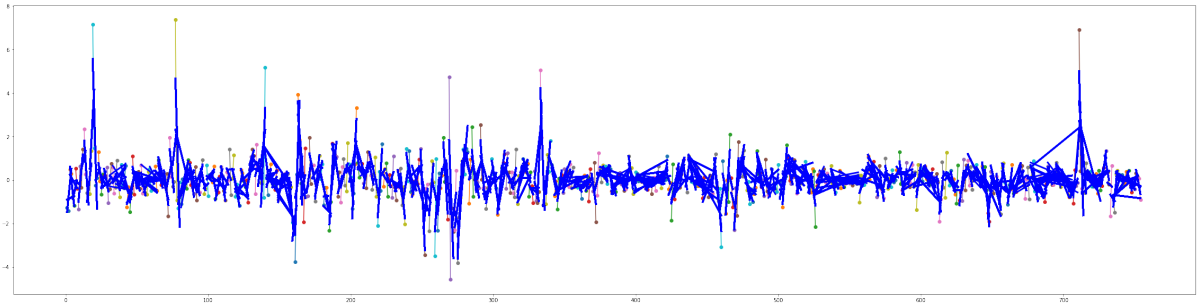


Figure 20. Amazon Segments with Regressions for Segments with MSE threshold 0.25.

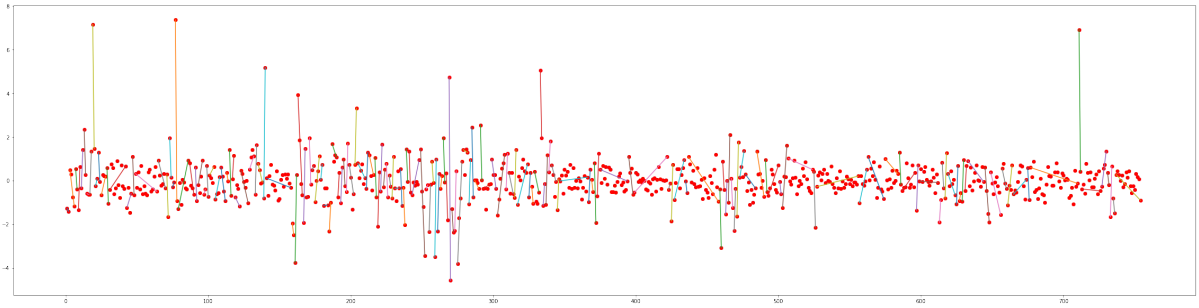


Figure 21. Amazon Segments after 208 Merging Iterations with MSE threshold 0.25.

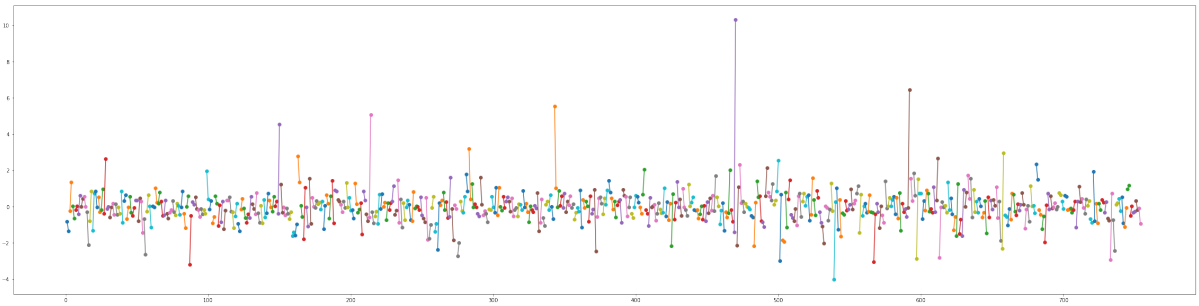


Figure 20. Nvidia Segments Before Merging.

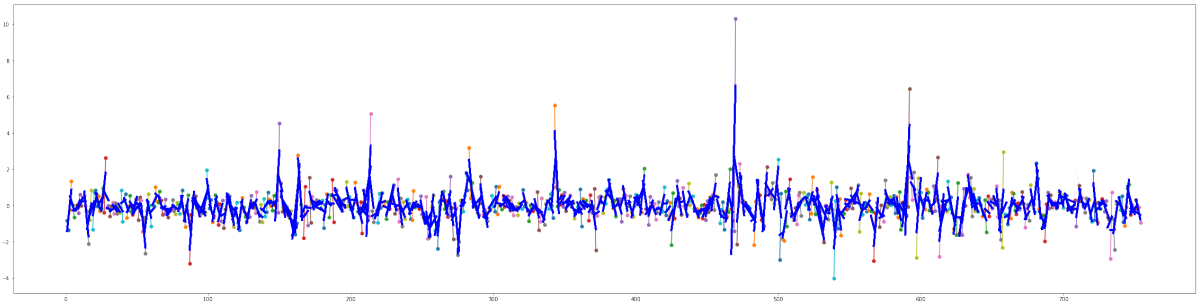


Figure 21. Nvidia Segments with Regressions for Segments with MSE threshold 0.04.

Figure 23. Nvidia Segments after 46 Merging Iterations with MSE threshold 0.04.

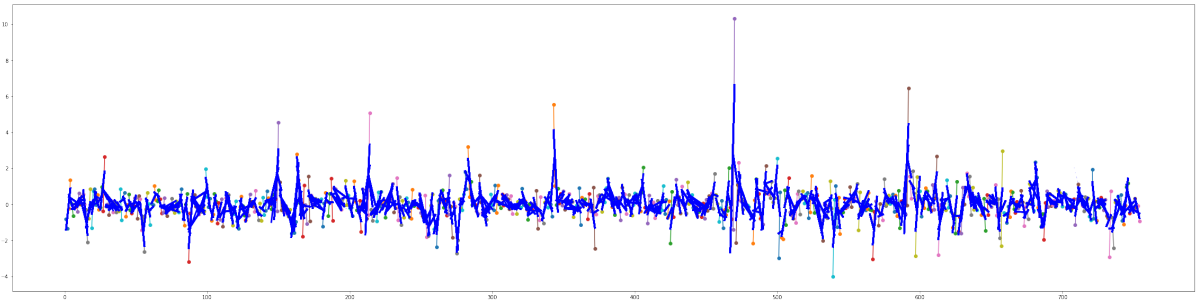


Figure 24. Nvidia Segments with Regressions for Segments with MSE threshold 0.1.

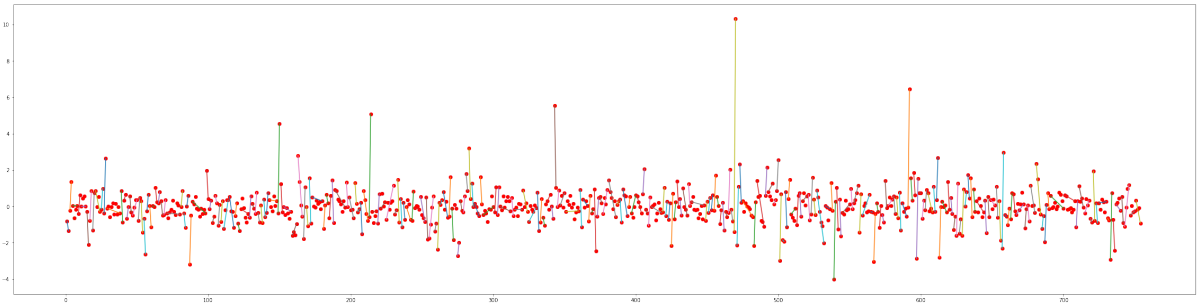


Figure 25. Nvidia Segments after 46 Merging Iterations with MSE threshold 0.1.

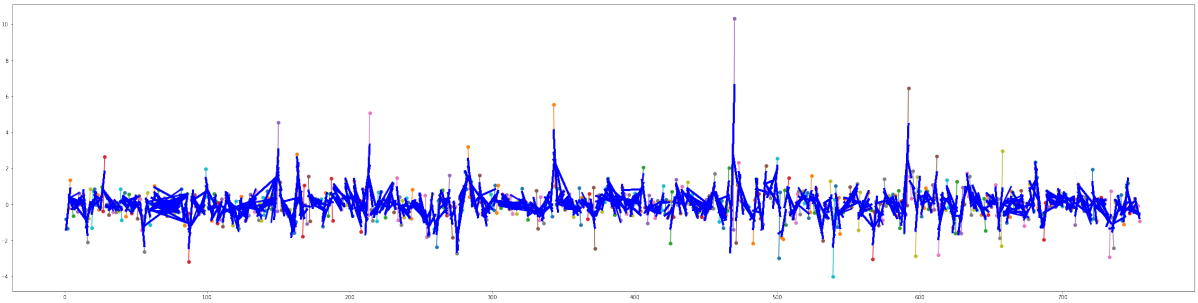


Figure 26. Nvidia Segments with Regressions for Segments with MSE threshold 0.25.

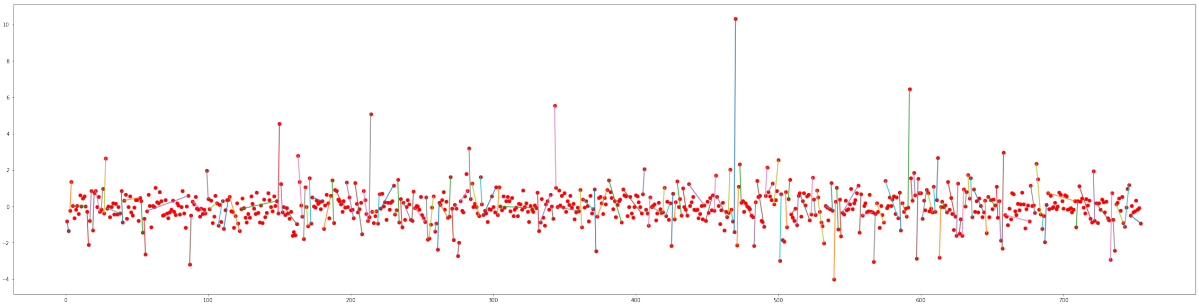


Figure 27. Nvidia Segments after 46 Merging Iterations with MSE threshold 0.25.

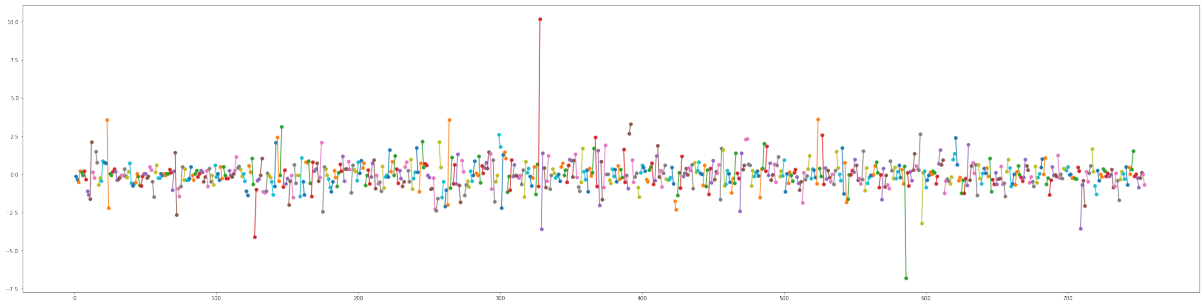


Figure 28. AMD Segments Before Merging.

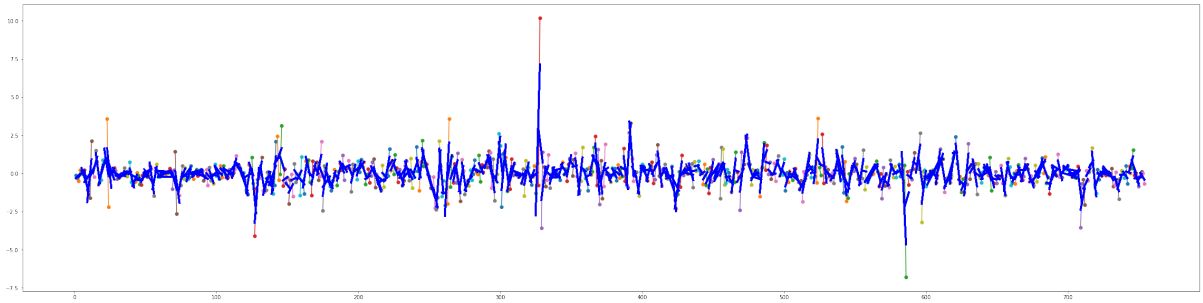


Figure 29. AMD Segments with Regressions for Segments with MSE threshold 0.04.

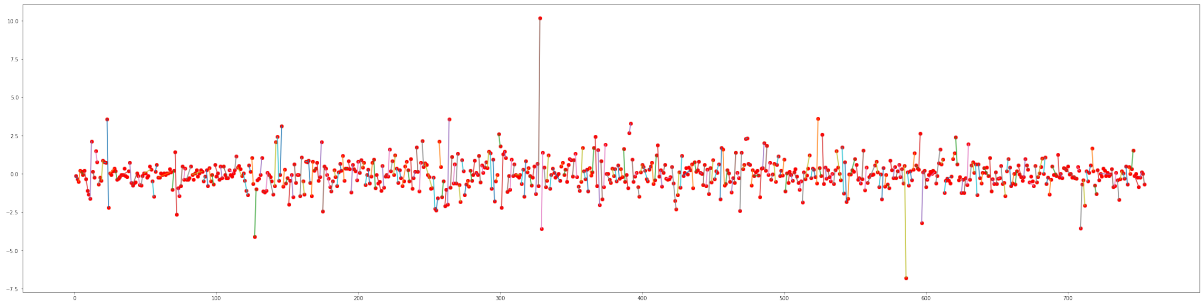


Figure 30. AMD Segments after 54 Merging Iterations with MSE threshold 0.1.

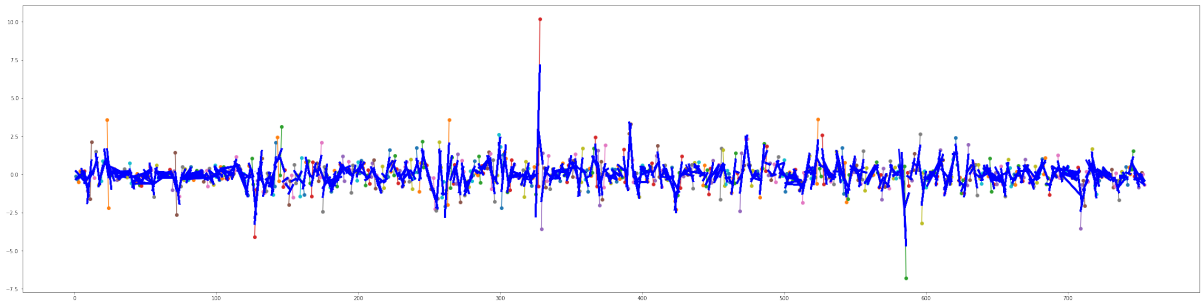


Figure 31. AMD Segments with Regressions for Segments with MSE threshold 0.1.

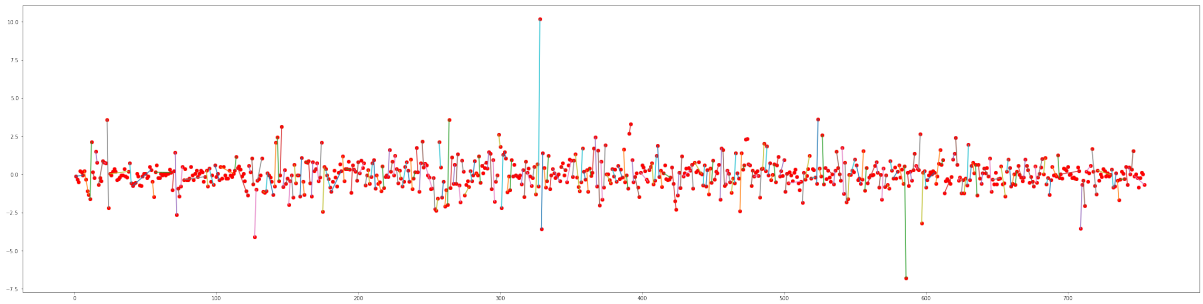


Figure 32. AMD Segments after 120 Merging Iterations with MSE threshold 0.1.

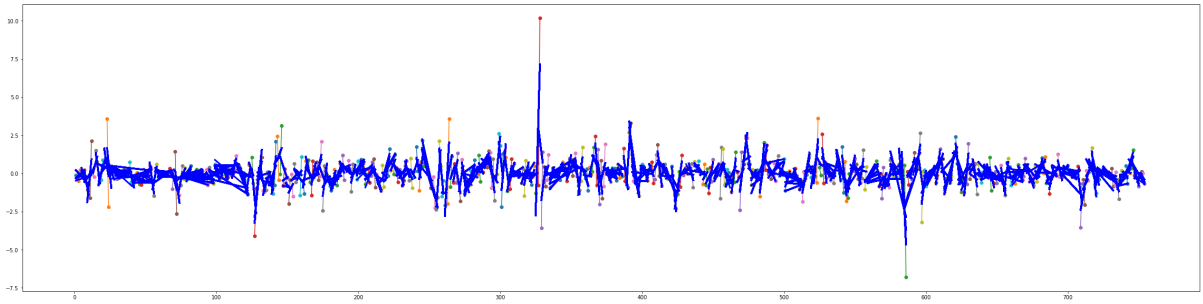


Figure 33. AMD Segments with Regressions for Segments with MSE threshold 0.25.

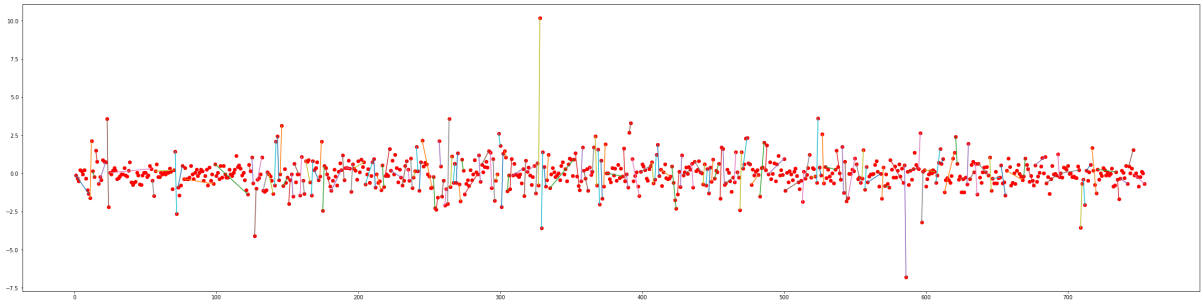


Figure 34. AMD Segments after 200 Merging Iterations with MSE threshold 0.25.

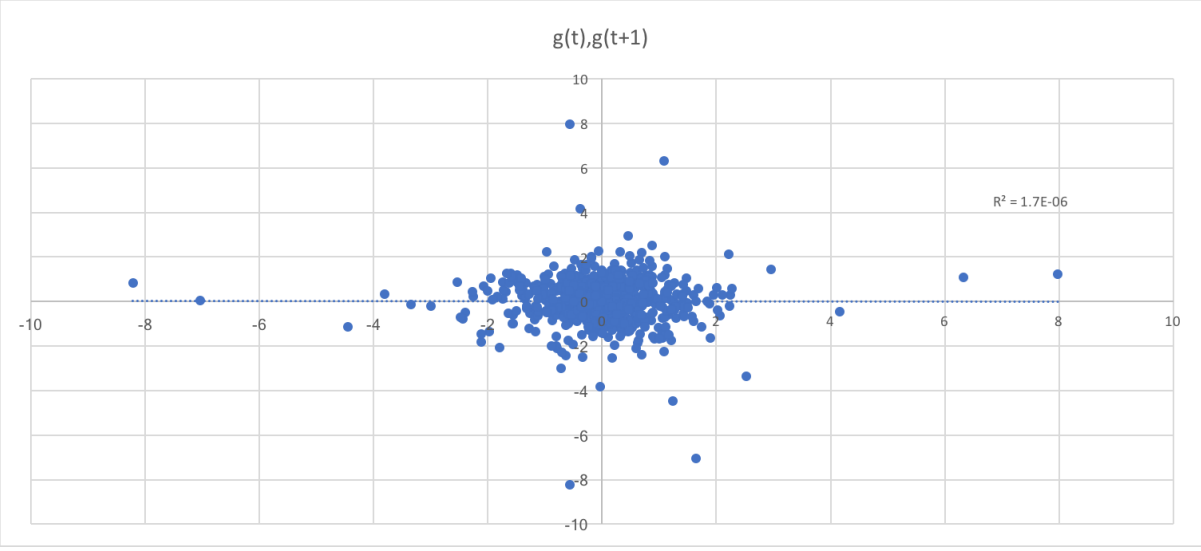
Task №4.

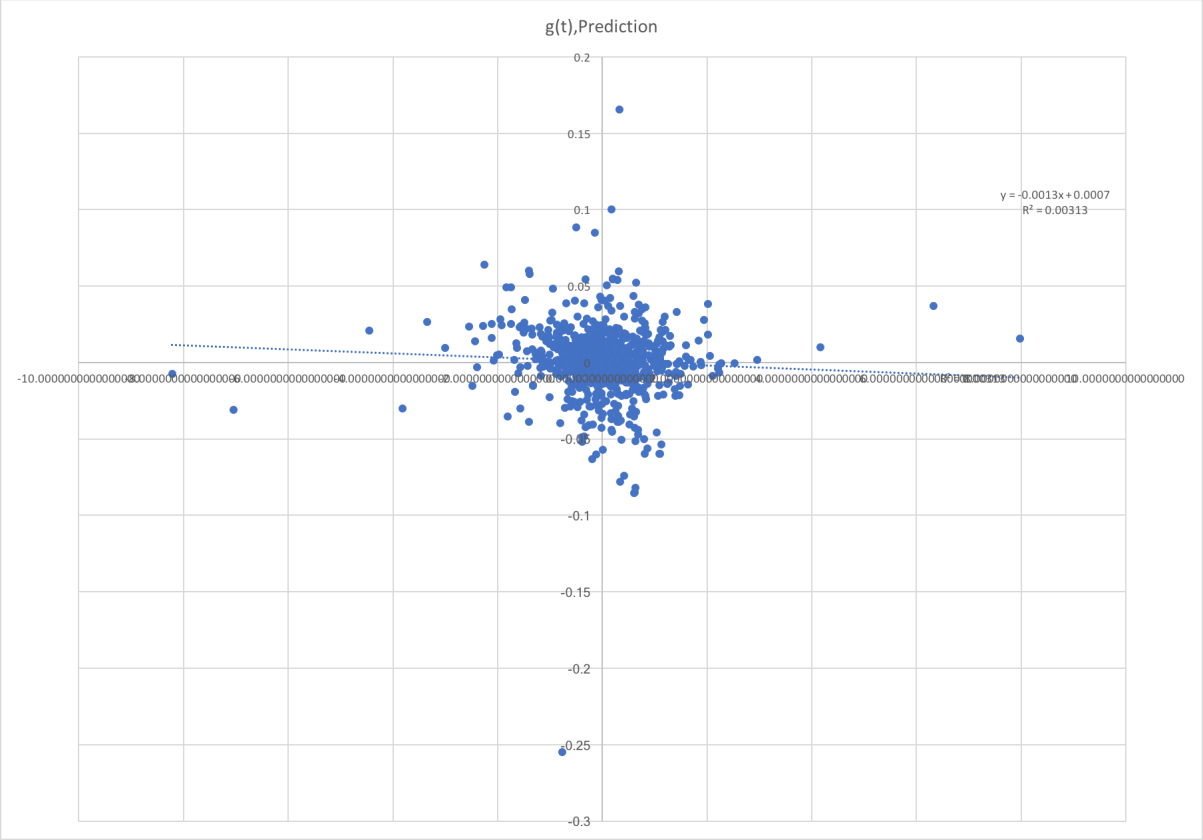
Prediction.

For this task I've chosen Ebay transformed and normalised data as a target $g(t)$ and 3 other datasets as a supporting data $d_1(t), d_2(t), d_3(t)$.$g(t+1)$ as a linear function:

$$\hat{g}(t+1) = \psi(g(t), d_1(t), d_2(t), d_3(t))$$

Plotting:





Coefficient of determination for next-day forecast prediction is 1.7×10^{-6} which is much less than one. So this mean that price cannot be predicted well .This model is not accurate(0.00017 percent).

Coefficient of determination for linear model prediction is 0.00313 which is not much less than one. So this indicates that price can be predicted better. This model is not really accurate as well (0.03 percent). [3]

References:

1)https://blackboard.le.ac.uk/bbcswebdav/pid-1474258-dt-content-rid-3889648_2/courses/MA4022/Preprocessing%281%29.pdf

2)https://en.wikipedia.org/wiki/Missing_data

3)http://stattrek.com/statistics/dictionary.aspx?definition=coefficient_of_determination