ANOMALY DETECTION IN A TIME SERIES DATA

INSTRUCTIONS AND PROBLEM STATEMENT: Given a data frame, in which the anomalies are pre-identified, create an algorithm, which would identify the anomaly and give the score for the identified anomaly. The algorithm should behave just like a human would give a score to the anomaly, that is:

- It should take the past anomalies into account, and just not give the output based on the deviation from the mean
- And, in case if the anomaly has a similar magnitude as compared to the previous anomalies, it should mimic human behavior.

OBJECTIVE:

To identify anomaly and give a score between 0 to 100, to the anomalies that have been identified, considering all the data until that given timestamp, and not looking at the data ahead of it.

ABSTRACT:

This report discusses the approaches that I applied, for solving the problem of giving a score to the anomaly, in case the anomaly is detected.

KEYWORDS:

Z-Score, Normalization

PROJECT OBJECTIVE:

The task is to devise an algorithm for scoring pre-identified anomalies in time series.

APPROACH:

1st Approach:

- Basically, the first approach was, to know how to predict the output as an anomaly, because if I were able to predict the point as anomaly, I could take out the confidence of that prediction, multiply it by 100, and that could work as an anomaly.
- So, for that, I had used three approaches:
    - Take into account only the **value** column
    - Take into account the **value** column, apply Normalization technique, outlier detection techniques, and some sampling techniques to make the data of anomalous and non-anomalous to be balanced
    - While selecting the algorithms, I had used the following algorithms:
        - KNN
        - LOGISTIC REGRESSION
        - XGBOOST
        - RANDOM FOREST
        - LSTM

The notebook and the code for the same can be found in here

THE SHORTCOMINGS:

- Basically, the main aim of the approach was to get an idea, about how an input value is predicted to be anomalous or non-anomalous, so that, I could take an input timestamp's value, and the weighted average of all the timestamp before it, and make the model outcome the prediction, and then return the confidence score for the same.
- However, the problem was, any technique or any model was not able to classify all the points which belonged to the anomalous category.
- And also, the weighted average thing was not clear to me, and maybe the ML algorithms behave according to their own assumptions, and not according to the requirements, which is not good in our case
- So, I thought of using the statistical and Normalization technique, which is my second approach to solve the problem of anomalous scores

Final Approach:

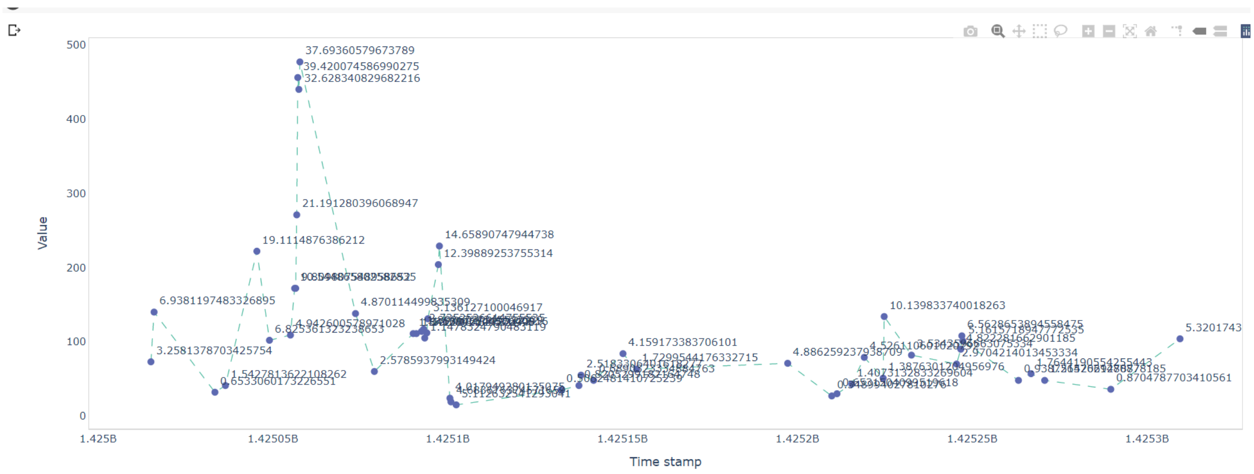
- The notebook for this approach can be found out here
- This time, I created a column known as deviation which would measure the deviation between the prediction and the value column, since this is the point where any human would give the score, means give the score based on the deviation observed between the predicted and the actual values
- In this method, I used Z-Score to find the outliers. And then found out that each of the anomalies can be figured out with the help of this technique, so I went forward with this technique.
- Now, my approach was suppose if I take the max value until the given timestamp T, and do some shifting scaling type of process, of all the data, and then compare it with the maximum deviation until now maybe that could be a potential candidate.

Why is the above method a potential candidate?

- If I were to be assigned to give a score for the anomaly, I would first consider the maximum deviation happened up till now, (this was my first step), and then scale all the points around the 0, and then take the ratio of given data point (scaled version), and the max data point (scaled version), and then output that score.
- Due to this, if the given data point has a deviation higher than the maximum, it would automatically get 100, and else wise, it will get a score comparable to the maximum deviated data point

Results:

As you can see from the below diagram, the outcomes are presented as desired, and hence, it proves that this method can be a potential candidate for the anomaly score.



Summary:

So, in order to summarize, what I did was, create a distance column, which would measure the difference between actual and predicted values, and then apply some normalization technique, and then compare it with the maximum deviation till now, and then output the score.

I think that more experimentation can be done, with the help of feature extractors like LSTM, which can be replaced with the Normalization technique, and possibly more combinations can be made to improve the results.