Forecasting the Closing Price of Amazon Stock

Week 3: Data Pretreatment

Github Repository: AMZN

The goals of this week's task are as follows:

- Checking for continuous measurements with the same frequency
- Checking for data synchronicity
- Checking for missing values
- Identifying and eliminating possible outliers using STL decomposition
- Conducting a mini-literature review (i) on dividing a long time-series into sub-samplings (multiple sequences). How can the sub-sequencing be done? How does seasonality affect sub-sequencing? Can we consider trends and seasonality in LSTM models? (ii) what are some typical standardisation methods for LSTM.

Checking for continuous measurements with the same frequency

The dataset comprises continuous measurements of the Open, High, Low and Close prices, along with the Volume of traded Amazon stock in the stock exchange market. Although the data is intended to be sampled uniformly (every business day), this ideal scenario is disrupted by the absence of trading on public holidays in the host country (USA). From the start date on January 3, 2006, to the conclusion on December 29, 2017, an anticipated 3129 business days were expected. However, the dataset has 3019 samples. To rectify the irregular sampling rate resulting from market closures on public holidays, we employ data imputation techniques. Specifically, the Last-Observation-Carried-Forward (LOCF) or fill-forward technique is deemed most appropriate. This method aligns with the nature of financial markets, where the absence of trading on a specific day implies that the current market value is equivalent to the value recorded on the preceding day.

Checking for data synchronicity

The data is synchronous across all variables, meaning that measurements for Open, High, Low, Close prices, and Volume of traded stock are consistently recorded at the same time steps. The data adheres to a regular and uniform sampling frequency, making it synchronous and ensuring that each variable is observed simultaneously.

Checking for missing values

Notably, there are no missing values in the dataset.

Identifying and eliminating possible outliers using STL decomposition

Assuming that the STL decomposition captures the trend and seasonality patterns correctly, it is possible to use the decomposition to identify possible outliers in the data. Whilst setting up the decomposition to be robust to some form of outliers, the residual plots in Figure 1 are obtained for the Opening, Closing, Highest, and Lowest Prices.

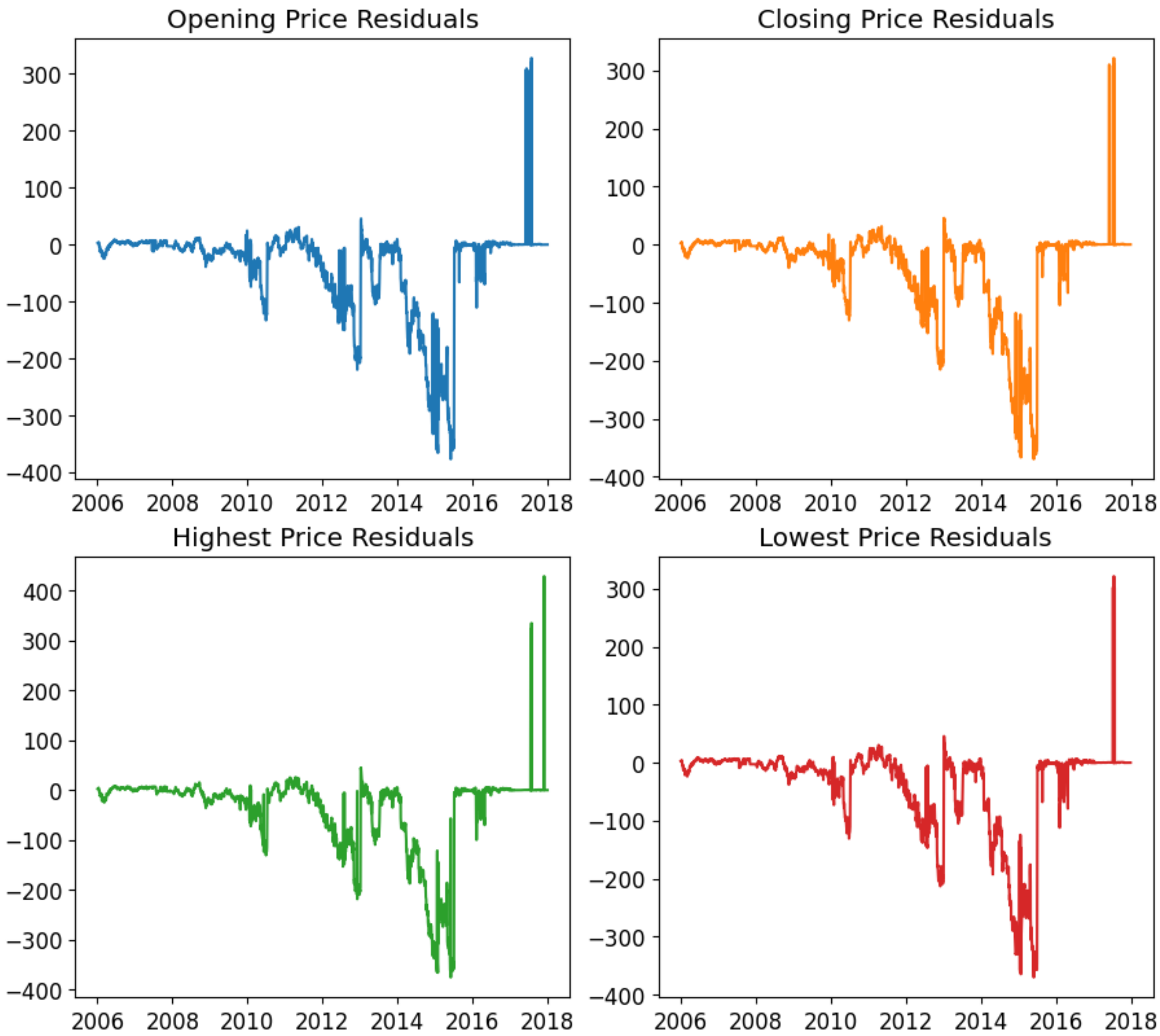

Figure 1: *Residual plots of the Open, High, Low and Close prices of Amazon Stock after STL decomposition.*

Using both global (based on mean and standard deviations of the whole set of residuals) and local (based on local deviation of the density of a given sample with respect to its neighbours), the spikes around the year 2017 in Figure 1 have been identified as possible outliers.

Even though action can be taken to deal with these potential outliers, simply replacing them without considering the cause of their occurrence is perilous. This is because these observations may provide useful information about the time series process under investigation - information which should not be disregarded during forecasting. Therefore, we have opted not to take any action at this stage.

Mini Literature Review

(i) *Dividing a Long Time-Series into Sub-Samplings:*

- *Sub-Sequencing Methods:* There are several methods for extracting sub-sequences from a longer sequence, each serving different purposes in time series analysis. Some common methods that are used and are to be considered in our analysis involve fixed-Length, and sliding windows among others. Fixed-length windows is where the time series is divided into equal-sized segments. For instance, if you have a time series of length N and choose a window size W, you would create $\frac{N}{W}$ sub-samples. On the other hand, sliding windows involve moving a fixed-size window through the time series with a specified step size. This results in overlapping or non-overlapping sub-sequences.

- *Impact of Seasonality:* Dividing a time series into sub-samples requires careful consideration of seasonality. One approach is to align windows with the seasonal pattern, facilitating the capture and preservation of seasonality in the sub-samples. Additionally, techniques such as seasonal decomposition of time series, such as STL decomposition, can be employed before sub-sequence extraction to effectively isolate and separate the seasonality component from the overall time series data.

- *Consideration of Trends and Seasonality in LSTM Models*: Recurrent Neural Networks (RNNs) face difficulties when dealing with lengthy sequences and issues related to numerical stability. In contrast, Long Short-Term Memory (LSTM) networks address these challenges by employing memory cells equipped with gates. These gates play a crucial role in regulating what information the cell remembers and produces as output (Lecture notes 10.b, LSTM equations). That's why LSTMs have the ability to capture and retain information about both short-term patterns (e.g., seasonality) and long-term trends in the time series.

(ii) Standardization Methods for LSTM

Normalisation/standardisation is commonly used in many data-driven models in the data preprocessing step to help ensure that the transformed data has certain desirable statistical properties [1]. For the case of deep neural networks, for which LSTM is an example, the most common standardisation techniques are:

- Min-Max Scaling

Min-max scaling involves transforming the data to a specific range, often between 0 and 1 or -1 and 1, whilst preserving the relationships between data points. For each observation for a particular feature, it is obtained as follows:

$$Scaled\,Value\ =\ \frac{Unscaled\,Value - Min}{Max - Min}\,(nMax\ -\ nMin)\ +\ nMin$$

where min and max represent the minimum and maximum value of the feature under consideration respectively. The lower and upper bounds to rescale the data are represented with nMax and nMin respectively [2].

- Mean Scaling

This method eliminates the data offset by deducting the mean of a feature from every instance of that particular feature [2]. The procedure is outlined as follows:

$$\widehat{x} = x - \mu$$

- Standardisation (Z-Score Normalisation)

In this case, we transform the data to have a mean of 0 and a standard deviation of 1. This is achieved by computing standard scores/ z-scores which are obtained by subtracting an observation x from the mean μ and dividing by standard deviation σ [2].

$$z = \frac{x - \mu}{\sigma}$$

Standardisation is useful when the features in the dataset have different scales or when the data distribution is expected to be normal. It helps to centre the data around zero and express values in terms of the number of standard deviations from the mean.

REFERENCES

[1] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. Normalization Techniques in Training DNNs: Methodology, Analysis and Application. IEEE Transactions on Pattern Analysis & Machine Intelligence, 45(08):10173–10196, 2023.

[2] Dalwinder Singh and Birmohan Singh. Investigating the Impact of Data Normalization on Classification Performance. Applied Soft Computing, 97:105524, 2020.