Assignment 1. Sequential Data Exercise

Environmental Time-Series

1. Decompose the water quality and one input variable of your choice into the three components. Plot them and discuss.

   For this part I decided to take the 1st column from Xdata and Ydata as a whole, we can see the results in Fig. 1 & Fig.2.

   From Fig.1 and Fig.2 we see some seasonality observed in the Xdata column 1. For Y data we observe a seasonality as well.

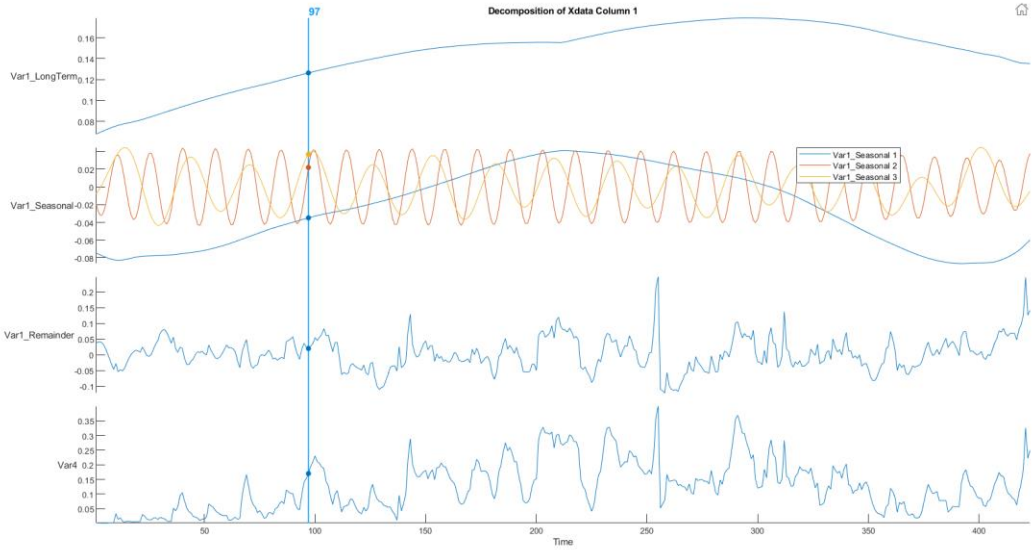

Fig.1

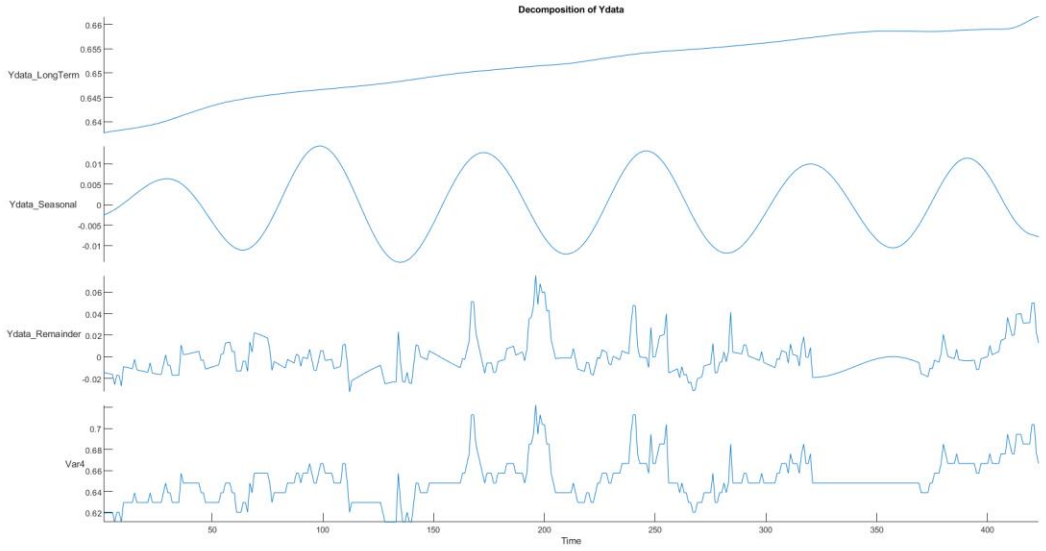

Fig.2

2. Test if there is weekly, monthly or quarterly seasonality.

   For checking different seasonality, we will divide and check if there is seasonality in 7 days, 30 days, 90 days.

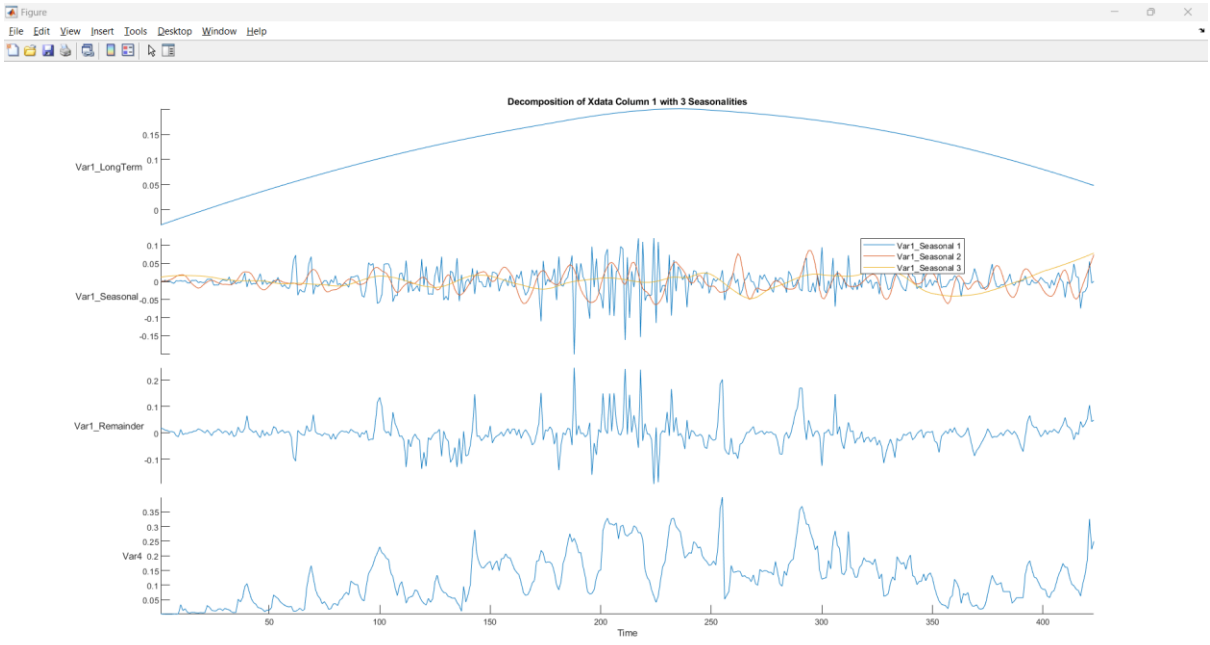

Fig. 3

From Fig. 3 we observe that weekly and monthly seasonality might be present in the xdata but quarterly seasonality is all over the place and there is no information to gather from the figure.

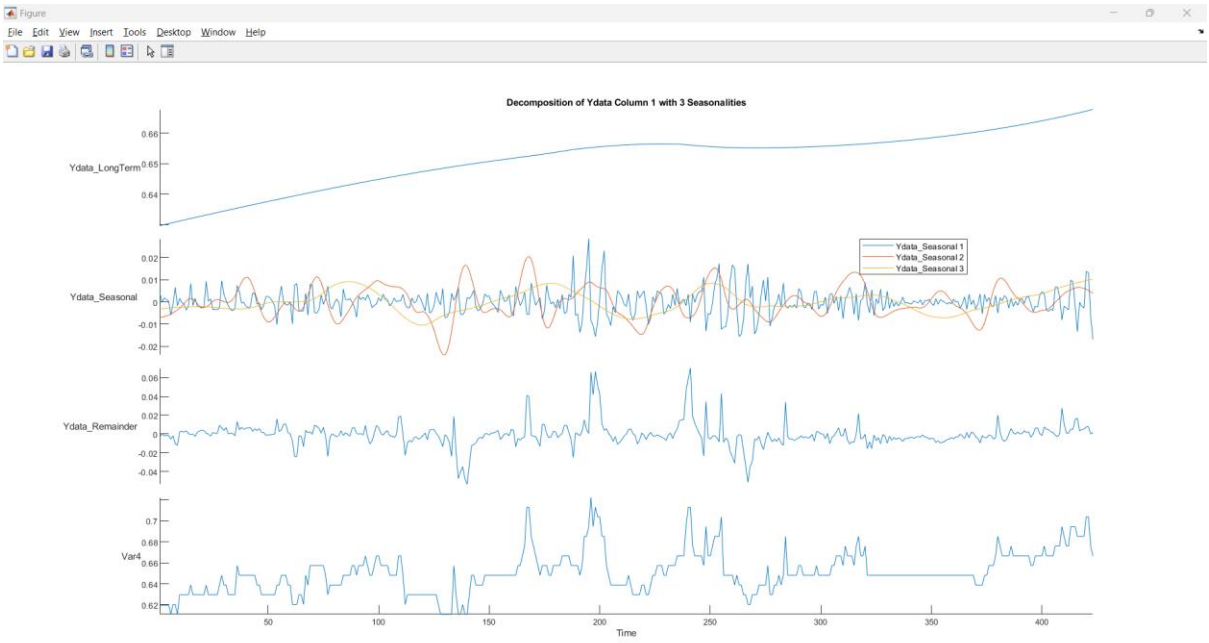

Fig.4

From Fig.4 Ydata exhibits seasonality in all three compartments i.e. Daily, Monthly, and Quarterly.

3. Perform k-means clustering on the water quality variable. Are you able to identify seasonality?

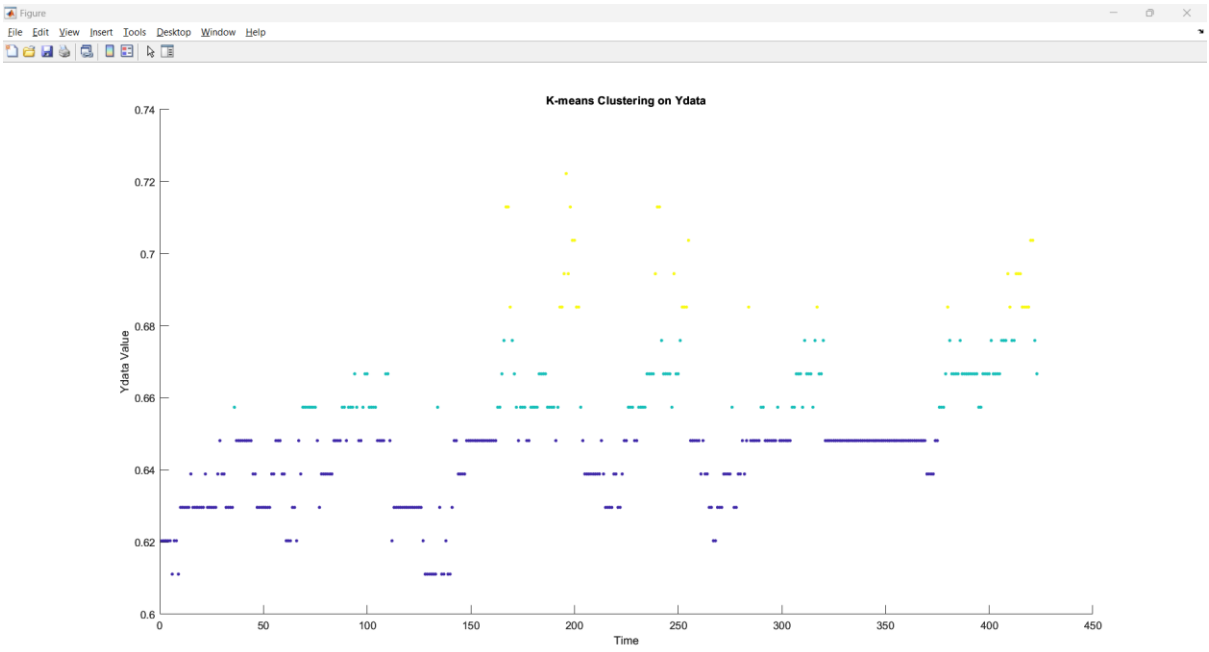

Fig.5

From Fig.5 we can divide the Water quality into 3 different clusters although a much more robust algorithm is required to further solidify this approach.

4. Perform k-means clustering on the multivariate time series. Can you identify seasonality?

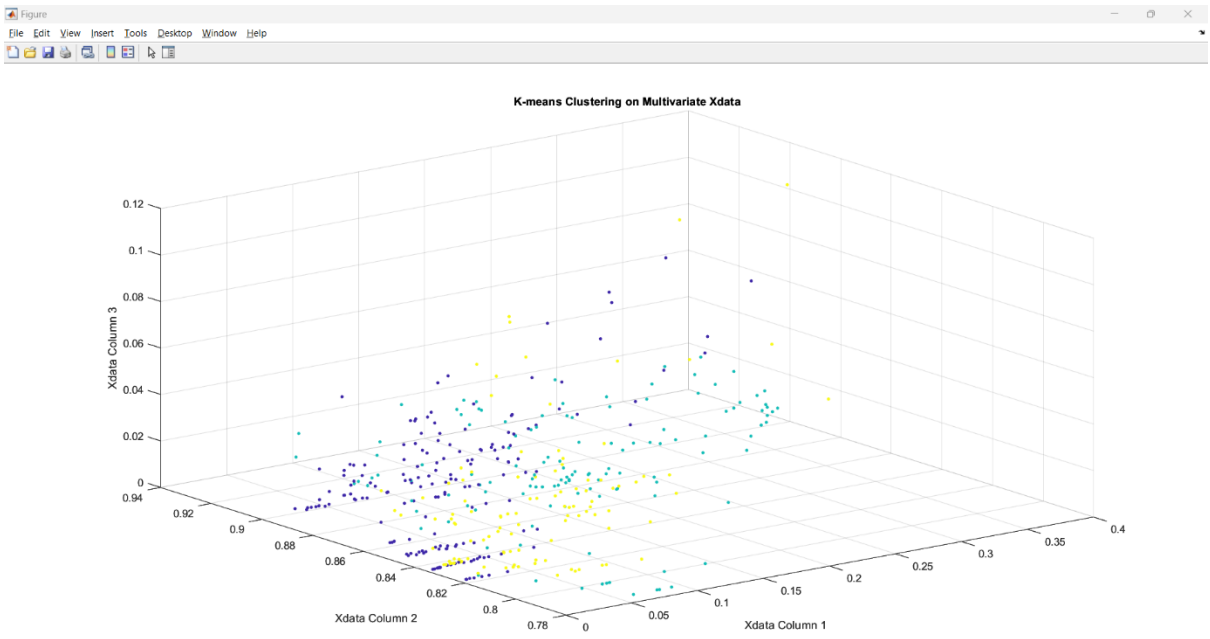

Fig.6

From Fig.6 we cannot identify a particular distinction between 3 clusters for time-series data.

5. How can this time-series data be divided into training and validation partitions, if you were to train a model for water quality predictions? Make a strategy

   Below are some of the strategies for training and validation partition of time-series data.

   a. Sequential Splitting:

      Allocate the first portion of your time series for training and the later portion for validation.

      For example, you can use the first 80% of the data for training and the remaining 20% for validation.

   b. Sequential Splitting:
      Allocate the first portion of your time series for training and the later portion for validation.
      For example, you can use the first 80% of the data for training and the remaining 20% for validation.

c. Sequential Splitting:
   Allocate the first portion of your time series for training and the later portion for validation.
   For example, you can use the first 80% of the data for training and the remaining 20% for validation.

Sequential Text Data

1. Load a text data of your choice. Make a plan on data pre-processing for analysis of the text componence.

   In the analysis of text data, preprocessing plays a crucial role due to the presence of numerous punctuations, prepositions, and irrelevant characters. Consequently, the initial step involves cleansing the data by removing insignificant characters. This process can be executed through the following sequential steps:

   - Tokenization: This entails breaking down the text into individual words or tokens.
   - Part-of-Speech Tagging: This aids in understanding the grammatical structure of the text.
   - Stop Words Removal: Common stop words, which add little meaningful information to the text, are eliminated.
   - Word Normalization: Words are normalized by lemmatizing them to their base form.
   - Punctuation Removal: Punctuation is typically devoid of significant meaning for text analysis.
   - Short Words Removal: This step aids in concentrating on more informative terms while eliminating most prepositions.
   - Long Words Removal: Excessively long words are excluded, as they might be outliers or errors.

2. Perform the pre-processing steps.

   This step is handled in the code file.

3. Plot a word cloud for (a) the cleared vocabulary (b) the 2N gram (c) the 3N gram

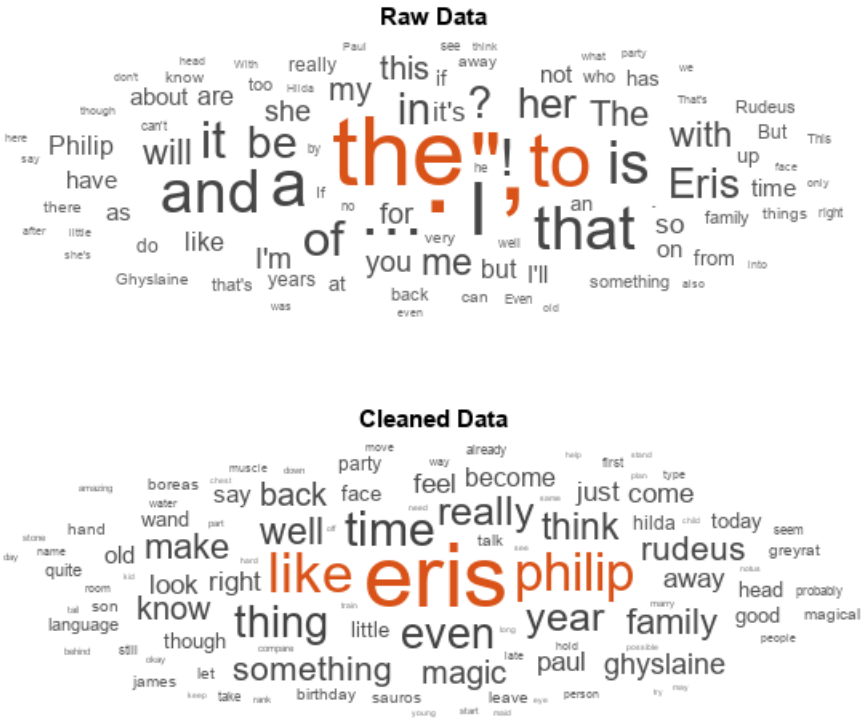

Fig.7

From Fig.7 we observe a lot of different words taking bigger chunk of the document in raw data but after cleaning it some words start to pop up like ‘eris’, ‘philip’ and ‘like’.

From Fig.8 we start to observe some clear combinations of words ‘even though’, ‘magical stone’, ‘year old are some of the combinations which are more prominent than others comparatively.

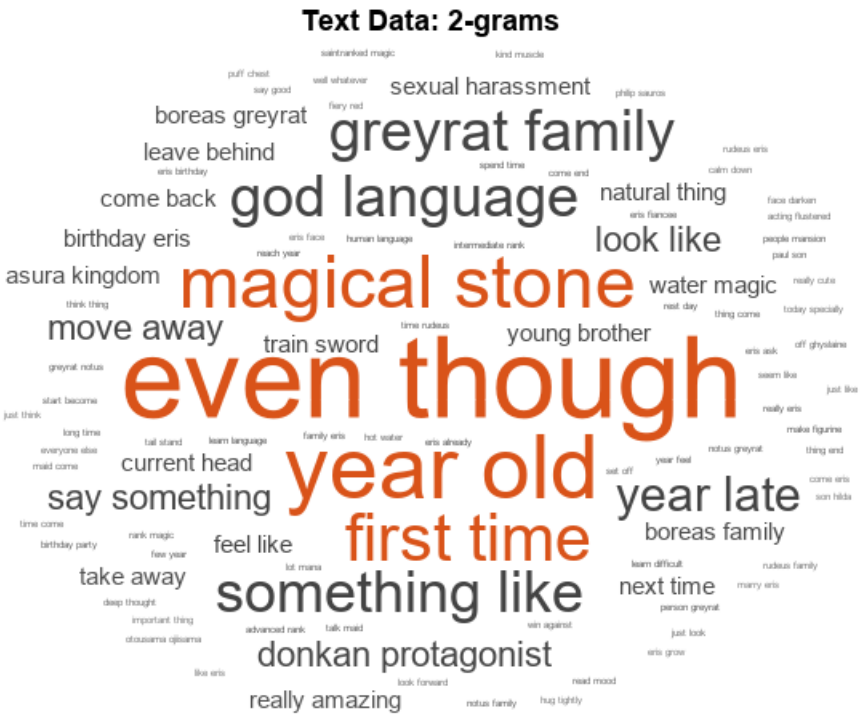
Text Data: 2-grams
sexual harassment
boreas greyrat
greyrat family
leave behind
come back
god language
natural thing
birthday eris
look like
asura kingdom
magical stone
water magic
move away
train sword
young brother
even though
year old
current head
say something
year late
first time
boreas family
feel like
take away
something like
next time
donkan protagonist
really amazing

Fig.8

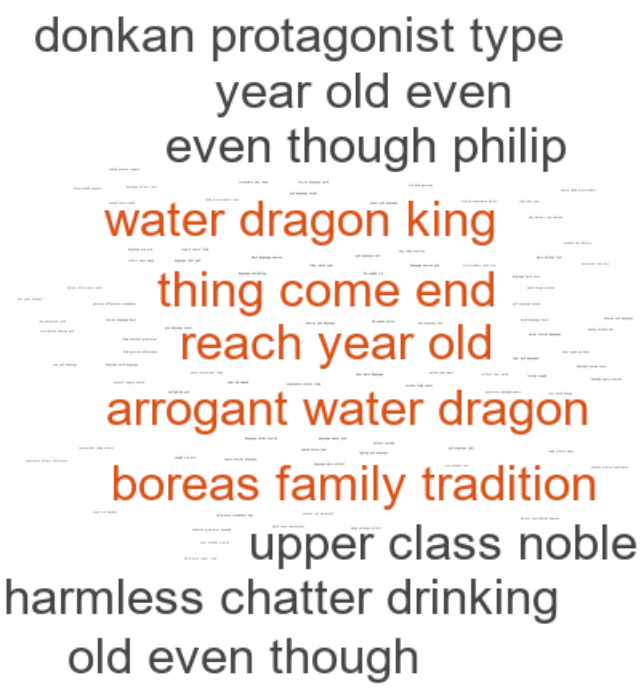
Text Data: 3-grams
donkan protagonist type
year old even
even though philip
water dragon king
thing come end
reach year old
arrogant water dragon
boreas family tradition
upper class noble
harmless chatter drinking
old even though

Fig.9

From Fig.9 we see that combinations such as ‘arrogant water dragon’, ‘boreas family tradition’, and ‘reach year old’ are more important compared to others.

4. Make a strategy for segmenting the text data into sub-sequences ready to be fed to a neural network. We assume the purpose of the network is predicting the next letter, in a text generation application. Write an example pair of input (X) - output data (Y) pair for your case.

When segmenting the text data for a text generation task, where the goal is to predict the next letter using a neural network, a common strategy is to create overlapping subsequence’s. Here's a general segmentation strategy:

Define Sequence Length:

Choose a fixed length for your input sequences. This determines how many previous characters the model will consider to predict the next one.

Create Overlapping Subsequences:

Create overlapping subsequences of the chosen length from the text data. Each subsequence represents a training example for the neural network.

Labeling:

For each input subsequence, the corresponding output (label) is the next character in the text.

One-Hot Encoding:

Encode both input and output sequences using one-hot encoding, where each character is represented as a binary vector.