Feature scaling is an essential preprocessing technique of input data for machine learning algorithms and beyond. Two widely used feature scaling techniques are *normalization* and *standardization*. For each of the following questions explain clearly and concisely:

(I) What is feature scaling?

Feature scaling is a technique for normalising the variety of independent variables or features in data. It is typically carried out during the data preprocessing step and is sometimes referred to as data normalisation in the context of data processing. We can look at an example: When we have some independent variables such as age, salary, and height, each of which has the corresponding ranges of (18-100 Years), (25,000-75,000 Euros), and (1-2 Meters), ‘feature scaling’ would enable all of them to fall within the same range, such as being centred around 0 or falling within the range (0,1), depending on the scaling method.

(II) Why scaling features of a dataset is necessary?

In various machine learning algorithms, we need to do scaling to bring all features of the data in a similar standing, such that so that one significant number does not impact the data model just because of their larger magnitude. Scaling can make a difference between a weak machine learning model and a strong one. Real-world datasets frequently include features that vary in size, scope, and units. We must therefore do feature scaling for machine learning models to comprehend these features on the same scale. Scaling is a ‘must’ in many algorithms when we want to get faster convergence, like the convergence in neural networks.

(III) What does normalization and standardization do to the data and the noise?

**Normalization:**

Normalization typically changes the values of the dataset’s numerical columns to a common scale without changing differences in the ranges of values. means rescaling the values into a range of [0,1]. In vector terminology, “normalizing” a vector most often means dividing by a norm of the vector and it often refers to rescaling by the minimum and range of the vector. This is done to ensure all the elements lie between 0 and 1 thus bringing all the values of numeric columns in the dataset to a common scale.

Example: Rescale feature to [0, 1]

***x′ = (x−min(x)) / (max(x)−min(x))***

Example: Rescale feature to [−1, 1]

***x′ = (x−mean(x))/ / (max(x)−min(x))***

**Standardization (Z-Score Normalization)**:

Standardization rescales the data to have a mean of 0 and a standard deviation of 1. This is called the norm for standardisation (unit variance). In the context of vectors, "standardising" a vector typically entails subtraction of the mean and then dividing by its SD (standard deviation). It basically rescales the features to unit-variance and zero-mean. so that they’ll have the properties of a standard normal distribution with

μ=0 and σ=1

where μ is the mean (average) and σ is the standard deviation from the mean.

***x′ = (x−μ) / σ***

Effect of normalization and standardization on noise:

Standardisation matches the variance of each variable. If that balance comes from relevant underlying processes relevant to what you are interested in, great. If the variance comes from noise, then you amplify the noise in variables with a lower information content.

Normalization and standardization take into consideration the noise in a data set to allow for more meaningful interpretation. Often, noise is represented in a data set by its standard deviation. If we scale the standard deviation to a constant like 1 by standardizing the data or if we limit the standard deviation by bounding the data when we normalize it, the noise is effectively accounted for.

2. We picked our time window to be 10 consecutive days.

After calculating the smoothened weeks, we observed that each week had 10071 data points corresponding to different time periods of the day. In order to calculate the ‘normal week’ we took the same time period data point from each of the 52 smoothened weeks and averaged it out. This resulted in the normal week with 10071 data points.   
In order to calculate the score we found the absolute difference between the each datapoint and their corresponding point in the normal week and then summed all the differences. Mathematically,

Score(B) =

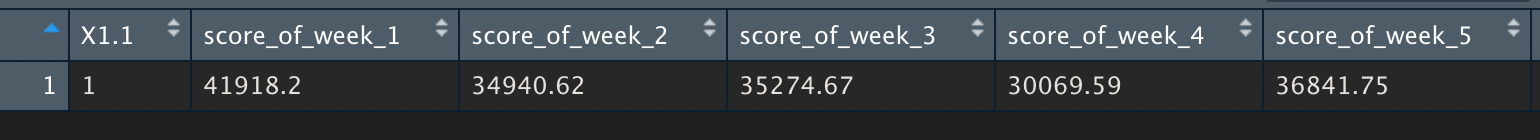
where A is the normal week and B is the smoothened week and Ai and Bi are the respective data points.

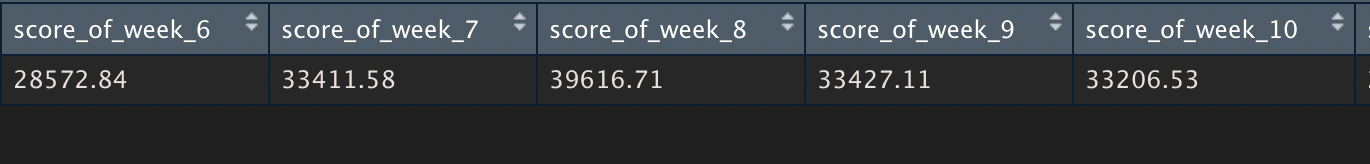
This score quantifies the difference between each data point in the normal and the smoothened week. The score is proportionally related to the anomalous nature of the week. Higher the score higher the anomolity. In order to find the most and the least anomalous week we simply found the minimum and maximum score respectively.

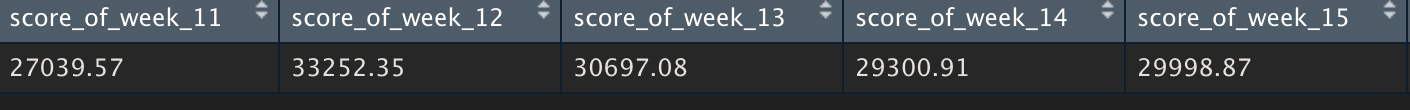
We also replaced all the NA values with 0 in order to avoid irregularities in the graph.

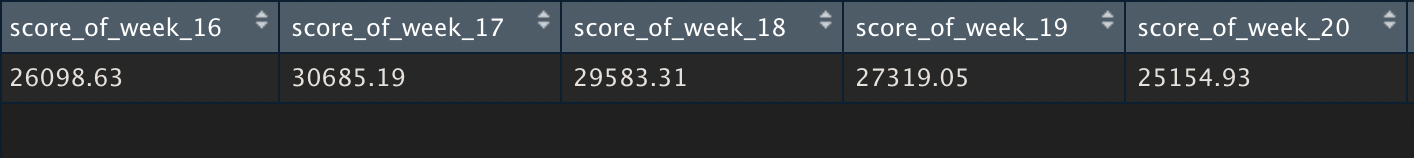
The most anomalous week is week 52 and the least anomalous week is week 37

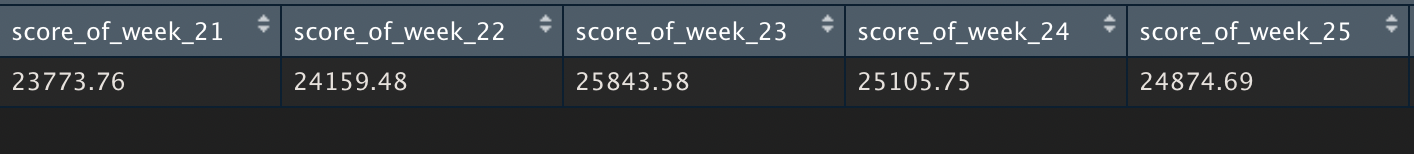
The scores are as follows:

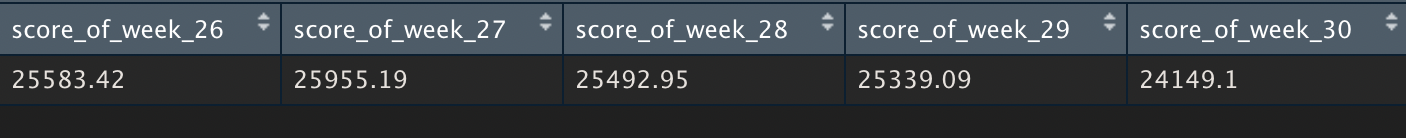


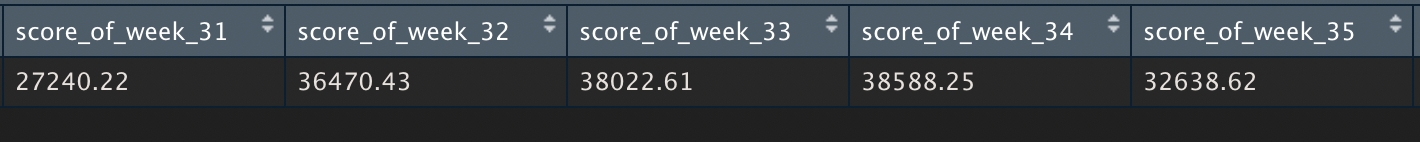


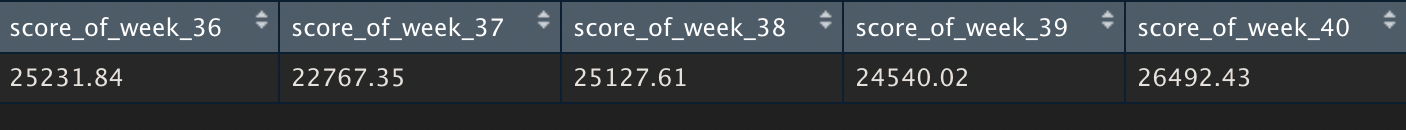


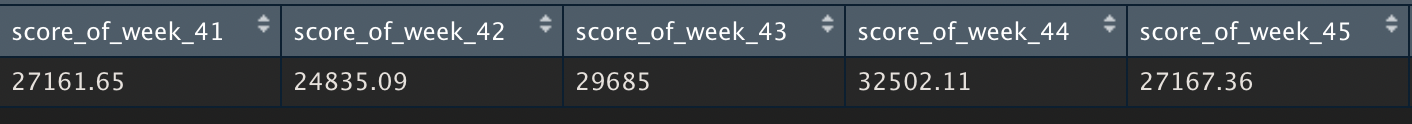
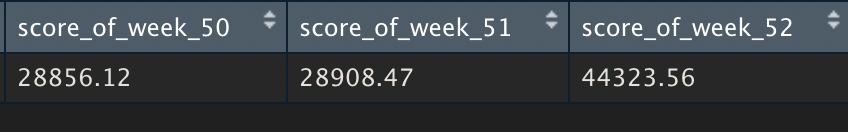
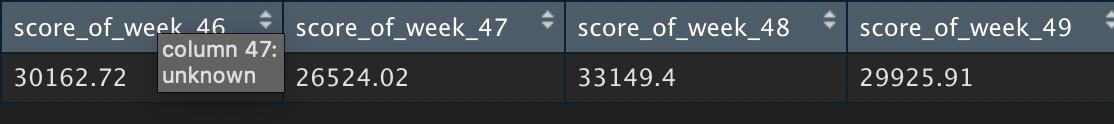


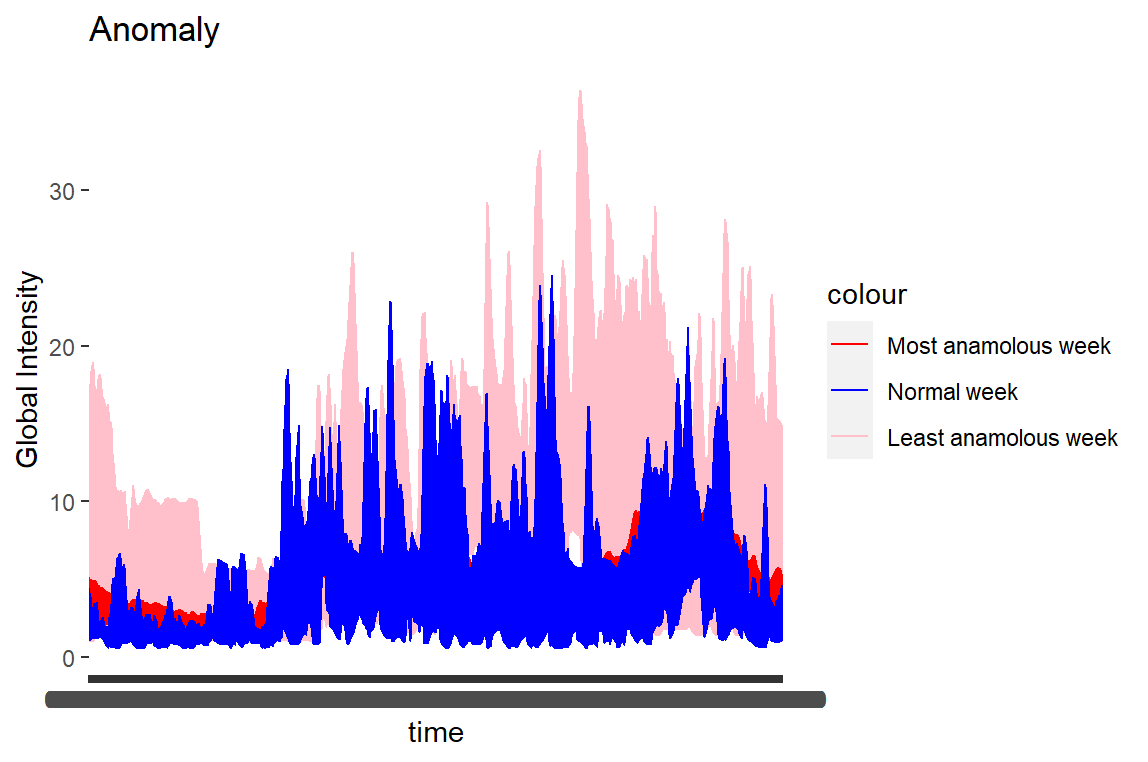










3.

Problem 1 explains how we efficiently calculate the exact probabilities of a sequence of observations given a certain number of states, i.e to find out which sequence of states maximizes the probability of the sequence of observations.

Mathematically,

O1, O2….. On is the sequence of observations

λ = (A, B, π) is the Hidden Markov Model provided where

A = State transition Probabilities i.e the probability of going from one state to another

B = Emission Probability i.e the probability of an observation given a specific state

π = Initial State probability i.e the probability of a state occurring independently

P(O| λ ) = Likelihood of the sequence of observations given the present Hidden Markov Model. P(O| λ ) = ∑P(O|Q).P(Q) , where Q is the assumed sequence of states.

For instance,

If the sequence of observations is O1, O2 and the assumed states are S1, S2

Then one possible sequence of state can be S2 and S1 . The probability in such a case would be:

P(O1O2| λ) = P(S2) \* P(O1|S2) \* P(S1|S2) \* P(O2|S1)

In case of an architecture like a supervisory control system observation would be the real time continuous data received from the supervisory system. For instance, the requests received by a server to access resources or in an office system it would be the computer activity or log in/out activities by employees. Possible states could be the system under attack or not under attack. Such observations form a multivariate time series that needs to be analyzed in (near) real-time. When these observations are received the conditional probabilities of the sequence of these observations is calculated and if it is extremely low then it can be flagged as an anomaly and countermeasures can be taken to contain the damage and facilitate cyber forensics.