CMPT 318
Group 19

Assignment 2

Q1.

What is feature scaling?

Feature scaling is one of the most important steps when it comes to dealing with data sets, and is usually done during the preprocessing of data. It is a method used to normalize data sets so that outliers don't have a strong impact due to the size of the value. There are many different types of scaling, the two most common being normalization and standardization. Scaling is most essential when machine learning algorithms are calculating distances between data.

Why is scaling features of a dataset necessary?

There are many benefits to feature scaling. One of those benefits is it alters the way the machine learning model operates. Prior to scaling, the machine learning model gives more importance to the numbers of higher values. When training the model, the larger values are weighed higher while the smaller values are weighed lower, without any regard for the units or what these numbers mean. There are things that humans can understand, but the machine only sees numbers and not the context. For example, there are two data sets with the weight of a product and its respective price. If the weight was in grams, the machine learning algorithm would put more importance on the numbers in the weight column due to how large they are. On the flip side, if the weight were to be converted to kilograms, the algorithm would heavily prioritize the data in the price column due to it being larger. Scaling will provide context for the algorithm and will adjust the variables to be within a certain range. Another reason for feature scaling has to do with the gradient descent. The gradient descent is an algorithm used to find the local minimum of certain functions, and a data set that has been scaled helps with the speed of convergence as opposed to a data set that has not been scaled.

What does normalization and standardization do to the data and the noise?

Normalization $X' = \dfrac{X - X_{min}}{X_{max} - X_{min}}$

Normalization, also known as Min-Max Scaling or Scaling Normalization, is one of the more simple methods and it consists of scaling the data to be between 0 and 1. Normalization is used when features are of different scales. For every set of data, the minimum becomes 0 and the maximum becomes 1. Normalization can be said to be situational, as outliers are not dealt with very well. Normalization also makes the data more concentrated around the mean and the noise is scaled to a small interval, meaning it does not handle noise well.

Standardization $X' = \dfrac{X - \mu}{\sigma}$

Standardization, also known as Z-Score Normalization, is a method of scaling that revolves around the mean with a unit standard deviation. The mean of attributes will be zero and the standard deviation will be 1. Unlike in normalization, standardization is not bounded to a certain range of numbers. The noise will not be affected by standardization so it is more robust to noises.

Q2.

Function blackBox2() is to get each weeks' score by calculating the square of the standard deviation. I chose to use the square of the standard deviation as the score because :
(1) it can enlarge the gap between an anomaly week and the normal week so any anomaly week will be more easy to detect.
(2) it can deal with negative gap and positive gap by squaring them.

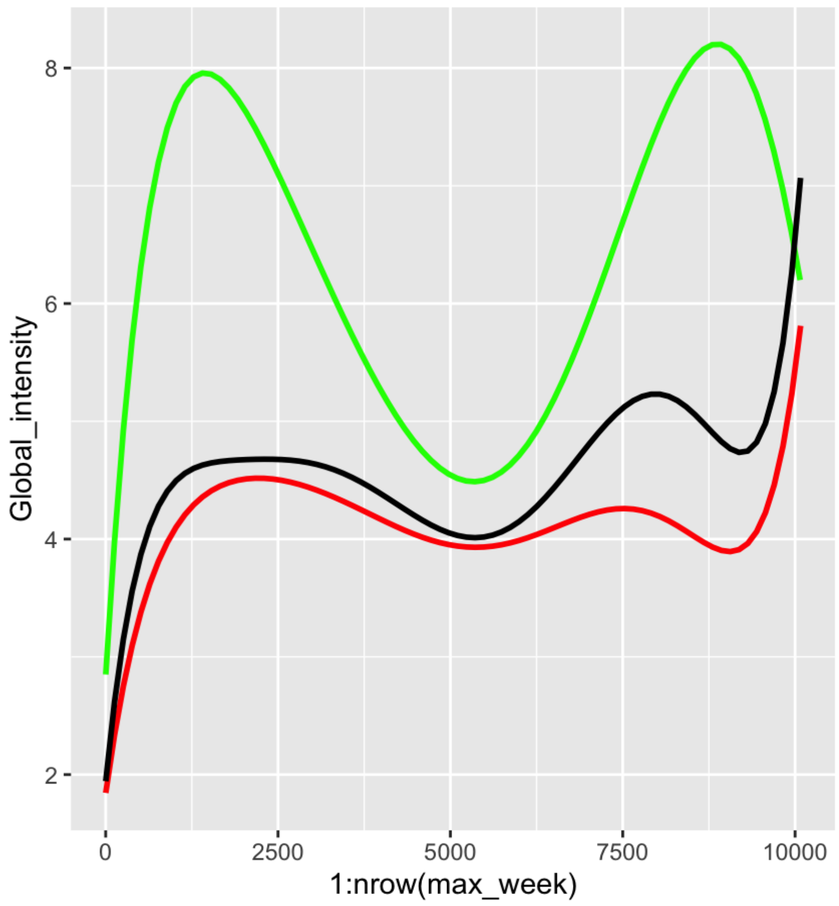
So the week has the highest score/the least score will be the most/least anomalous week, respectively. In our case, week 5 is the most anomalous week and week 39 is the least anomalous week.

The following is the anomaly score table for all weeks:

```
Week   Score
1    2 19.95157
2    3 17.13161
3    4  19.1526
4    5 21.47951
5    6 14.82127
6    7 12.43994
7    8 14.71533
8    9 13.58285
9   10 20.07717
10   11 14.48012
11   12 14.20593
12   13 14.50152
13   14 10.39268
14   15 17.33778
15   16 13.07962
16   17 10.39522
17   18 14.86736
18   19 9.821892
19   20 10.80992
20   21  8.40032
21   22 8.240968
22   23  8.34292
23   24 20.08858
24   25 8.812702
25   26 8.882951
26   27 8.236217
27   28 10.96945
28   29 12.22598
29   30 11.95951
30   31 13.71516
31   32 14.37513
32   33 13.78618
```

```
33   34 10.43034
34   35 9.430938
35   36 8.127391
36   37 9.302098
37   38 10.14661
38   39 7.843988
39   40 13.41919
40   41 12.64951
41   42 18.04648
42   43 15.78641
43   44 12.26261
44   45 21.15103
45   46 17.71145
46   47 18.60767
47   48 20.74426
48   49 13.18448
49   50 14.60919
50   51 15.01139
51   52 14.77499
```

The following is the plot of the smoothened versions of the most and the least anomalous weeks against the average smoothened week (Green line is the most anomaly week, red line is the least anomaly week and the black line is the normal week):



Q3.

The evaluation problem of HMM is summed up as calculating the probability that a model will produce a certain sequence of observations. This probability can be used to arrange a set of models from the highest probability of the model producing this sequence of observations to the lowest probability.

$$O = O_1, O_2, \ldots, O_T$$

Is the sequence of observations

$$P(O \mid \lambda)$$

Is the probability that from a model a certain sequence of observations O occurs.

In the context of anomaly detection, if you consider your initial sequence of observations to be the normal behavior of a certain operating system that you are going to train on. And then you create a set of models based on this normal behavior.

Then the highest $P(O \mid \lambda)$ represents the model that was most probable to achieve that normal behavior. The model obtained can be your trained model for anomaly detection.
The supervisory control system can test the trained model against new streams of data to see the probability that the normal behavior would produce this sequence. If the probability is sufficiently low then an anomaly possibly occurred.