Assignment-1 due 6.11.2020 11:59pm

Question 1. Anomaly Detection: Nelson Rules for Control Chart

Western Electric Company Rules (WECO) have been widely used for Shewhart control charts in order to increase the sensitivity of detecting assignable causes of process change.

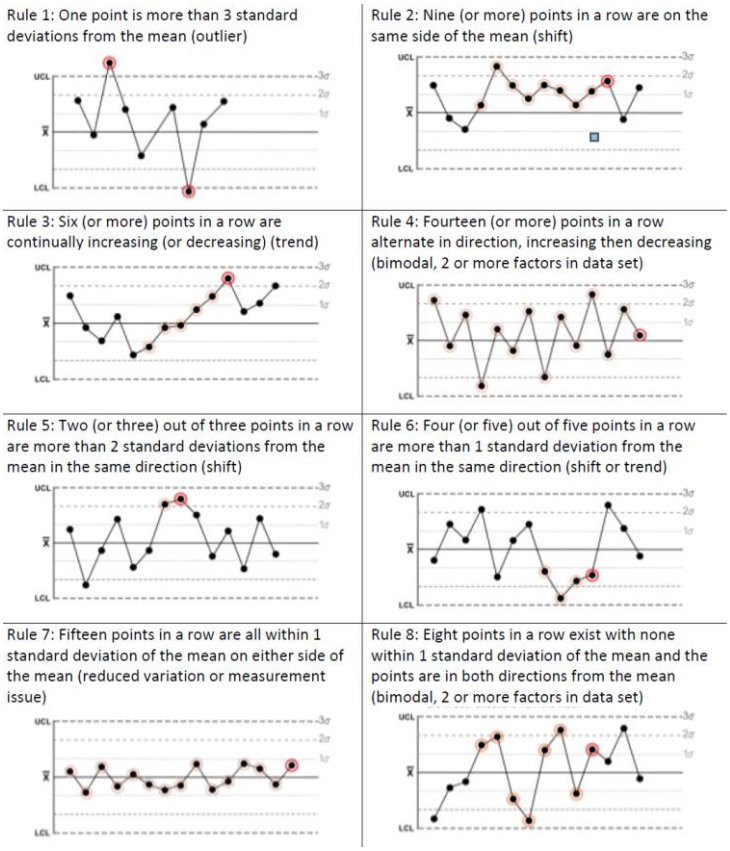
A 1984 update by Lloyd S. Nelson to the popular Western Electric Rules, in order to make the probability of detecting an out of control condition by chance approximately equal across all tests. The Western Electric Rules have probabilities that have more variability from one another (some are more likely than others to occur by chance). There are eight different Nelson Rules that are explained on below.

Implement a Python script that checks anomaly in daily credit card expenditure dataset (*Assignment1_Question1_data.xlsx*) according to average weekly expenses (The expenditures are in today's dollar, and 7 days=1 week). This script uses the first 52 weeks in the training (for calculating $\bar{x}$ and σ), so rest for testing.

Generate a table as csv file that should be as follows:

Week	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
52								
...								
156								

The code should fill the cell by 1, if there is an anomaly for the respected week and rule; otherwise, the value of the cell should be 0.



Question 2. Naïve Bayes Learning

Implement the solution for only one of the options provided below:

Option-1:

Implement a spam filter using Naïve Bayes learning on Enron emails dataset. You can download it from *http://www2.aueb.gr/users/ion/data/enron-spam*. Download only Enron1, and Enron6. Then, unzip the files and create a folder; each text file is an email; spam and ham go in separate folders.

You should implement the Naïve Bayes learning algorithm described in class you can use any Naïve Bayes package already implemented, i.e. *sklearn*, *nltk*. You must write a preprocessor that converts an email message into a vector of feature values (words), and then encode the features.

Split dataset 70% and 30% for training/testing. Train two different models; first trained from Enron1, and second from Enron6. Report confusion matrix and your prediction accuracy for each test set for both models.

Option-2:

Consider the weather problem discussed in the class. Implement your own Naïve Bayes classifier, that is you cannot use any Naïve Bayes package already implemented, i.e. *sklearn*, *nltk*. You must implement your own methods to calculate probabilities, store probabilities and predict the outcome for each record in the data set. You may use the data file used in the class, which is available through LMS.

Split dataset 70% (training data) and 30%(test data). Report confusion matrix and your prediction accuracy for each train and test set seperately.