Assignment #2

A REPORT ON

Multivariate Time Series Analytics (MVTS)

BY

1. Akshit Khanna 2017A7PS0023P
2. Vitthal Bhandari 2017A7PS0136P

Prepared in partial fulfilment of the course
Foundations of Data Science - CS F320

SUBMITTED TO

Dr. Navneet Goyal

Professor

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Task #1 - Identification, Implementation and Application of an MVTS Regression Algorithm

1. Identification of an MVTS regression algorithm

   We will use one of the most commonly used methods for multivariate time series regression – ***Vector Auto Regression*** (VAR). In a VAR model, each variable is a linear function of the past values of itself and the past values of all the other variables.

   Vector Autoregression (VAR) model is an extension of univariate autoregression model to multivariate time series data. A VAR model is a multi-equation system where all the variables are treated as endogenous (dependent). There is one equation for each variable as a dependent variable.
   A typical AR(p) model equation looks something like this:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t$$

   where α is the intercept, a constant and β_1, β_2 till β_p are the coefficients of the lags of Y till order p. Order 'p' means, up to p-lags of Y is used and they are the predictors in the equation. The ε_t is the error, which is considered as white noise.

   As for univariate distributed lag models, one should think carefully about which variables to include in a VAR, as adding unrelated variables reduces the forecast accuracy by increasing the estimation error. This is particularly important because the number of parameters to be estimated grows quadratically in response to the number of variables modeled by the VAR.

2. Implementation of ***VAR***

   The entire VAR model has been executed according to following steps in R. 'vars' library is used for the VAR model. The procedure to build a VAR forecasting model involves the following steps:

   a. Analyze the time series characteristics
   b. Test for causation amongst the time series (Using Multiple R Squared)
   c. Transform the series to make it stationary, if needed
   d. Find optimal order (p)
   e. Prepare training and test datasets
   f. Train the model

g. Roll back the transformations, if any.

h. Evaluate the model using test set

i. Forecast to future

3. Application of ***VAR***

One plausible application could entail *predicting future Air Quality Index* (AQI) values to make informed decisions to tackle the problems of air pollution. Using our dataset of Air Quality we can predict the future air quality index values based on the past data.

- Data Set used : Air Quality Data Set is used for the regression task. Dataset Link
  - Data Set Characteristics : Multivariate, Time-Series
  - Number of Instances : 9358
  - Number of Attributes : 15
  - Attribute Characteristics : real
  - Associated Tasks : regression

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level,within an Italian city. Data were recorded from March 2004 to February 2005 (one year) representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NO_x) and Nitrogen Dioxide (NO_2) and were provided by a co-located reference certified analyzer. Missing values are tagged with a -200 value.

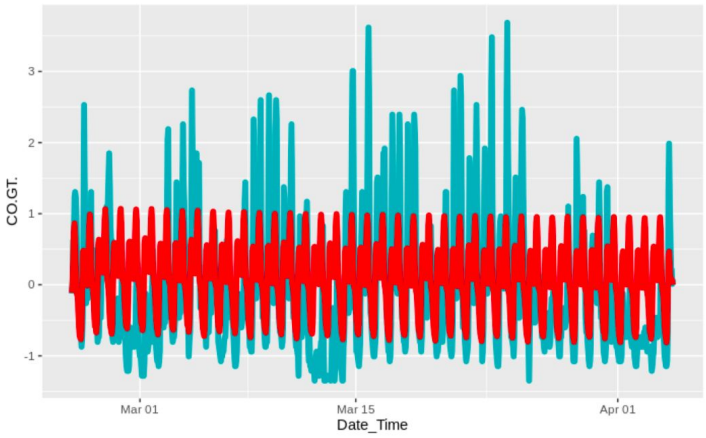
Missing values were replaced with the closest recorded values. 'NMHC.GT.' Class was removed from the data due to excessive missing values. The dataset is standardized with a mean of 0 and standard deviation of 1. Seasonality is also observed in the dataset with a rough frequency of 24 hours.
The dataset is divided into a 90 percent training set and 10 percent validation set to check the forecasting.

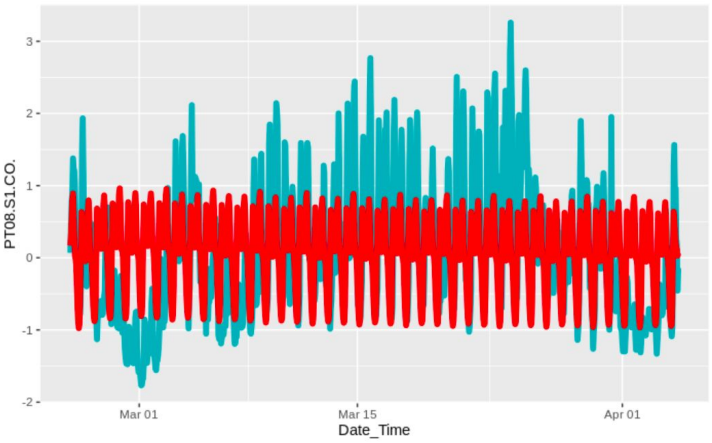
Number of lags in VAR Model - 5
The increase in the number of lags did not make any significant difference to the results.
Seasonal Frequency - 72

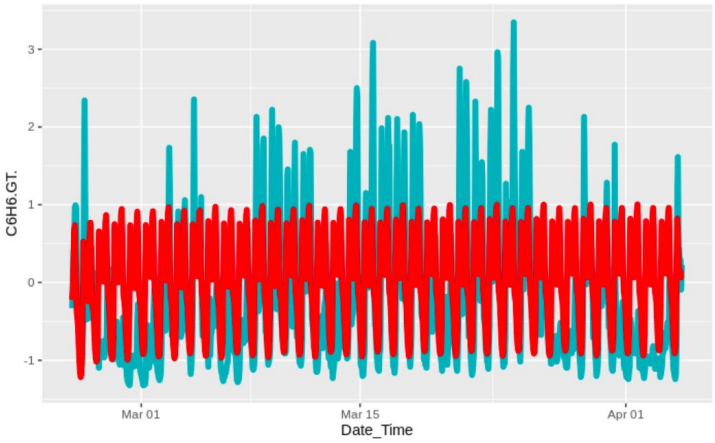
Attribute	Multiple R Squared Value (Adjusted)
CO.GT.	0.8339
PT08.S1.CO.	0.8691
C6H6.GT.	0.8269
PT08.S2.NMHC.	0.8642
NOx.GT.	0.8828
PT08.S3.NOx.	0.8902
NO2.GT.	0.8773
PT08.S5.O3.	0.8895
T	0.9873
RH	0.9553
AH	0.9866

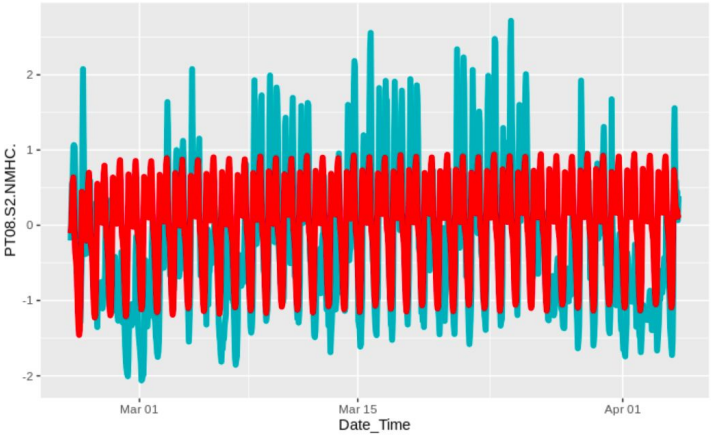
Table showing various attributes of the AQI dataset with their corresponding Multiple R squared adjusted values

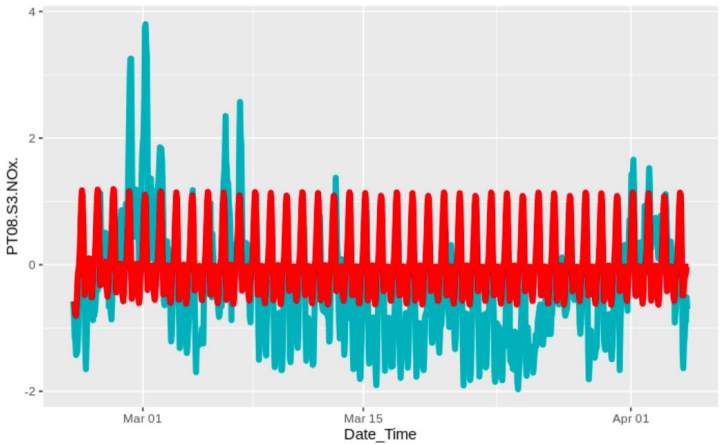
Given below on the next page are figures showing various plots. Each figure represents the plot (in green color) of value vs date/time of one of the attributes of the dataset. This way a plot has been created for all the attributes. Each of the figures also contains an overlapping plot (in red color) depicting the forecasted values as predicted by VAR.

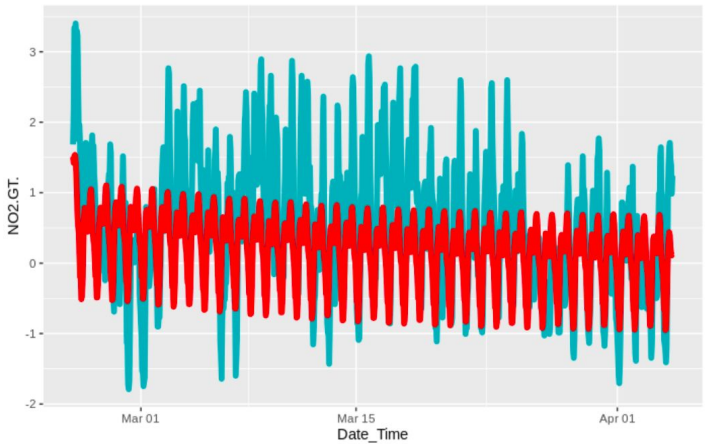

CO.GT.
3
2
1
0
-1
Mar 01
Mar 15
Apr 01
Date_Time

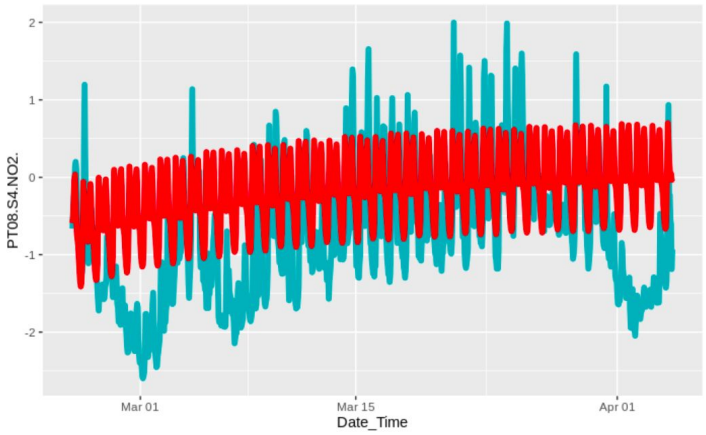

PT08.S1.CO.
3
2
1
0
-1
-2
Mar 01
Mar 15
Apr 01
Date_Time

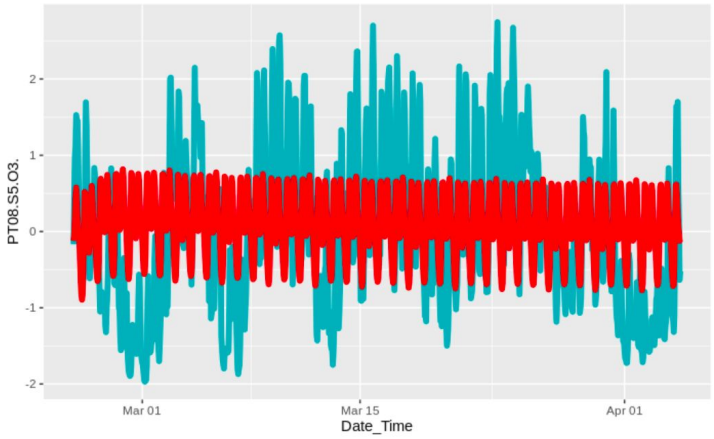

C6H6.GT.
3
2
1
0
-1
Mar 01
Mar 15
Apr 01
Date_Time

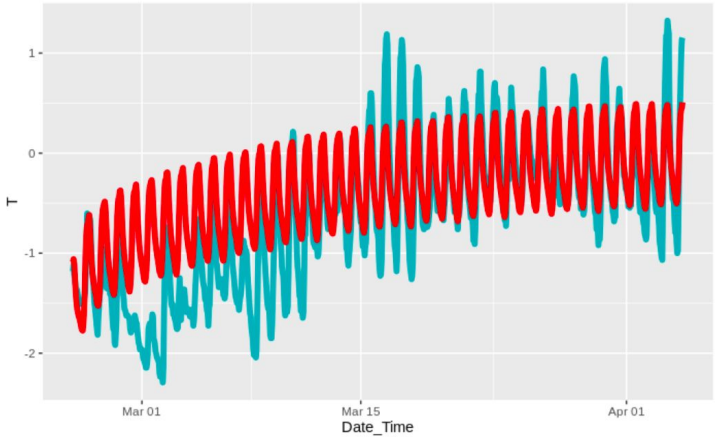

PT08.S2.NMHC.
2
1
0
-1
-2
Mar 01
Mar 15
Apr 01
Date_Time

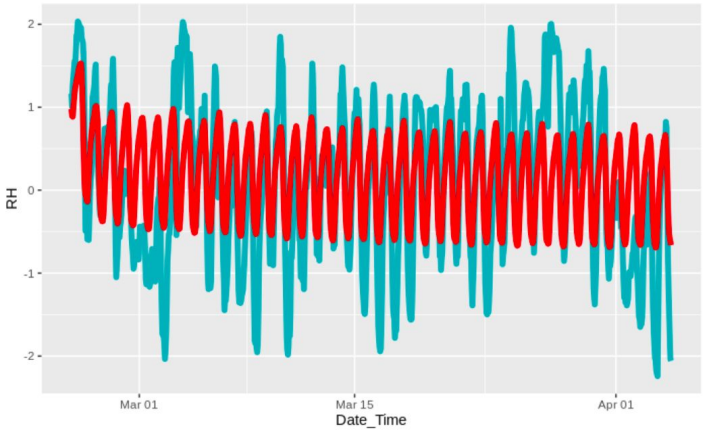

PT08.S3.NOx.
4
2
0
-2
Mar 01
Mar 15
Apr 01
Date_Time

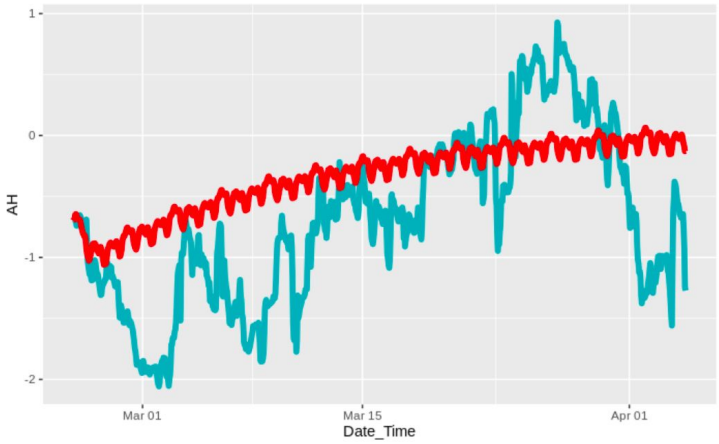

NO2.GT.
3
2
1
0
-1
-2
Mar 01
Mar 15
Apr 01
Date_Time












```
CO.GT. : 0.9089985
PT08.S1.CO. : 0.9160229
C6H6.GT. : 0.8529305
PT08.S2.NMHC. : 0.9464494
NOx.GT. : 0.8761507
PT08.S3.NOx. : 0.916845
NO2.GT. : 1.38062
PT08.S4.NO2. : 0.8279102
PT08.S5.O3. : 1.02734
T : 0.8699131
RH : 1.33301
AH : 0.6950206
```

Calculated RMSE values for the forecast

The above 11 figures show the plot of values of each attribute of the data set w.r.t time and the corresponding predicted value according to VAR

Task #2 - Identification, Implementation and Application of an MVTS Classification Algorithm

1. Identification of an MVTS classification algorithm

In a typical classification problem we are given a set of input features and a set of discrete output classes and we want to model the relationship between the two. But what happens with time series data is that the input features are not independent. In this case SVMs and Naive Bayes would not be a good choice since they assume that the input features are independent. The ***k-NN algorithm*** could still work however it relies on the notion of a similarity measure between input examples.

The ***Euclidean distance*** between two time series Q and C of length n is defined as :

$$d(Q,C) = \sqrt{\sum_{i=1}^{n}[Q(i) - C(i)]^2}$$

With a good similarity measure, small changes in two time series should result in small changes in their similarity. With respect to Euclidean distance this is true for changes in the y-axis, but it is not true for changes in the time axis (i.e. compression and stretching). It often produces pessimistic similarity measures when it encounters distortion in the time axis. The way to deal with this is to use dynamic time warping.

Dynamic time warping finds the optimal non-linear alignment between two time series. The Euclidean distances between alignments are then much less susceptible to pessimistic similarity measurements due to distortion in the time axis. There is a price to pay for this, however, because dynamic time warping is quadratic in the length of the time series used.

The first thing we do is construct an $n \times n$ matrix whose i,j^{th} element is the Euclidean distance between qi and ci. We want to find a path through this matrix that minimizes the cumulative distance. The optimal path is found via dynamic programming, specifically the following recursive function :

$$\gamma(i,j) = d(q_i, c_j) + min(\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1))$$

It should be noted that it is possible for one point in a time series to be mapped to multiple points in the other time series.

Due to its ***quadratic time complexity***, performing dynamic time warping multiple times on long time series data can be prohibitively expensive. However, there are a couple of ways to ***speed things up***. The first is to enforce a ***locality constraint***. This works under the assumption that it is unlikely for q^i and c^j to be matched if i and j are too far apart. The threshold is determined by a window size w. This way, only mappings within

this window are considered which speeds up the inner loop. Another way to speed things up is to use the ***LB Keogh*** lower bound of dynamic time warping. It is defined as :

$$LBKeogh(Q,C) = \sum_{i=1}^{n}(ci-Ui)^2 I(ci > Ui) + (ci-Li)^2 I(ci < Li)$$

Where Ui and Li are upper and lower bounds for time series Q which are defined as $Ui = max(q_{i-r} : q_{i+r})$ and $Li = min(q_{i-r} : q_{i+r})$ for a reach r and $I(\cdot)$ is the indicator function.

The LB Keogh lower bound method is linear whereas dynamic time warping is quadratic in complexity which make it very advantageous for searching over large sets of time series.

2. Implementation of ***k-NN*** with ***DTW*** and ***LB Keogh lower bound method***

The entire classification algorithm has been implemented from scratch in R.

1-NN algorithm uses dynamic time warping Euclidean distance. In this algorithm, for every time series in the test set, a search must be performed through all points in the training set so that the most similar point is found.

Given that dynamic time warping is quadratic, this can be very computationally expensive. We can speed up classification using the LB Keogh lower bound. Computing LB Keogh is much less expensive than performing dynamic time warping. And since $LBKeogh(Q,C) \leq DTW(Q,C)$, we can eliminate time series that cannot possibly be more similar than the current most similar time series. In this way we are eliminating many unnecessary DTW computations.

KNN algorithm is implemented directly with DTW only.

3. Application of the proposed MVTS classification algorithm

An interesting time series classification problem is predicting whether a subject's eyes are open or closed based only on their brain wave data (EEG). The problem was described and data collected by Oliver Rosler and David Suendermann for their 2013 paper titled " A First Step towards Eye State Prediction Using EEG "

- Data Set used : EEG Eye State Data Set is used for the classification task. Dataset Link
    - Data Set Characteristics : Multivariate, Sequential, Time-Series
    - Number of Instances : 14980
    - Number of Attributes : 15
    - Attribute Characteristics : integer, real

- Associated Tasks : classification

Specifically, an electroencephalography (EEG) recording was made of a single person for 117 seconds (just under two minutes) while the subject opened and closed their eyes, which was recorded via a video camera. The open/closed state was then recorded against each time step in the EEG trace manually.

The EEG was recorded using a Emotiv EEG Neuroheadset, resulting in 14 traces.

The output variable is binary, meaning that this is a two-class classification problem.

The dataset has been divided into sets of 50 observations as a time series with the majority class taken as the ground truth to classify the time series. The entire dataset is not used to decrease time taken to compute results.

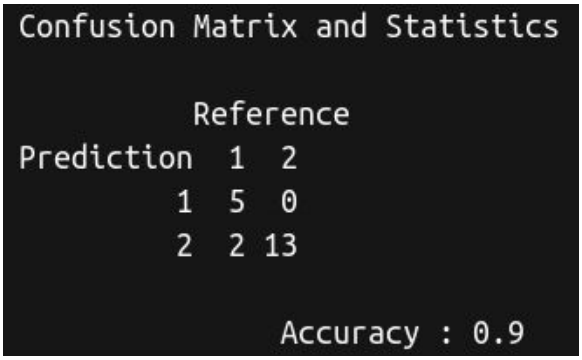
Training Set - 180 Time Series of 50 Observations
Test Set - 20 Time Series of 50 Observations

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1  5  2
         2  2 11

               Accuracy : 0.8
```

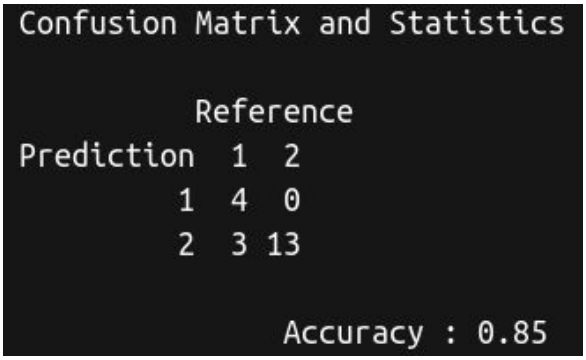
k = 1

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2
         1  5  2
         2  2 11

               Accuracy : 0.8
```

k = 3



k = 5



k = 7

Figures showing accuracy of classification prediction (via a confusion matrix) for different values of k (k=1, 3, 5 and 7)

Task #3 - Identification, Implementation and Application of an MVTS Clustering Algorithm

1. Identification of an MVTS clustering algorithm

   The algorithm with DTW and LB Keough lower bound method proposed for classification can also be applied to ***k-means clustering***. In this algorithm, the number of clusters is set apriori and similar time series are clustered together. Basically two steps are required for k-means clustering :

   a. Assigning data points to clusters

   b. Recalculating centroids of clusters

   The above two steps are carried out in a loop for a fixed number of iterations (until we are sure that the value of centroids doesn't change significantly anymore).

   As part of each iteration step (a.) requires that each time series of the dataset is compared to every other centroid and it is assigned to the centroid cluster which is closest (or most similar) to it. Again, similarity is calculated using Dynamic Time Warping Euclidean Distance. The speed of computing similarity is significantly sped up using a locality constraint and LB Keough lower bound method.

   Step (b.) is carried out once step (a.) is completed in which we recompute the value of centroids according to the newly formed clusters by taking the mean of each time series in that cluster. These new centroids act as seed for step (a) in the next iteration.

2. Implementation of ***k-means*** with ***DTW*** and ***LB Keogh lower bound method***

   The entire classification algorithm has been implemented from scratch in R.

   The number of clusters will be set beforehand and similar time series will be clustered together.

3. Application of the proposed MVTS clustering algorithm

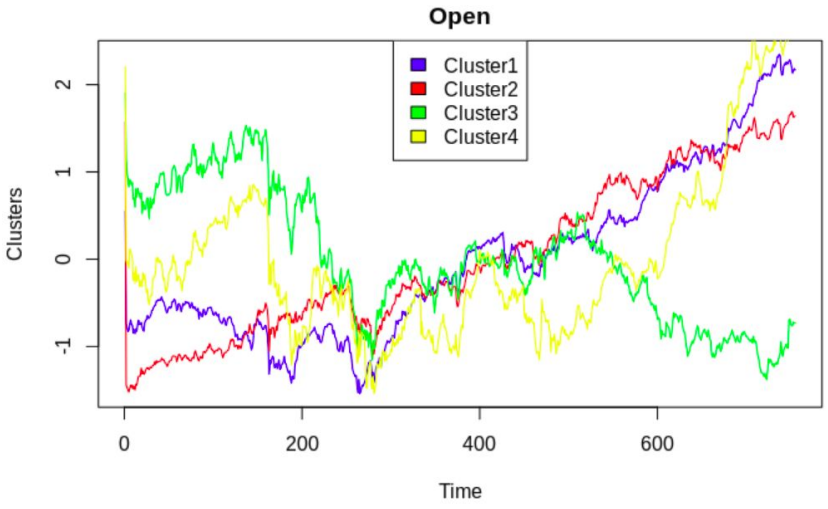
   The proposed clustering algorithm for MVTS can be used to group together companies on the S&P 500 index (Standard and Poor's 500) based on their historical stock prices data. This way comparisons can be made among similar companies in the same cluster.

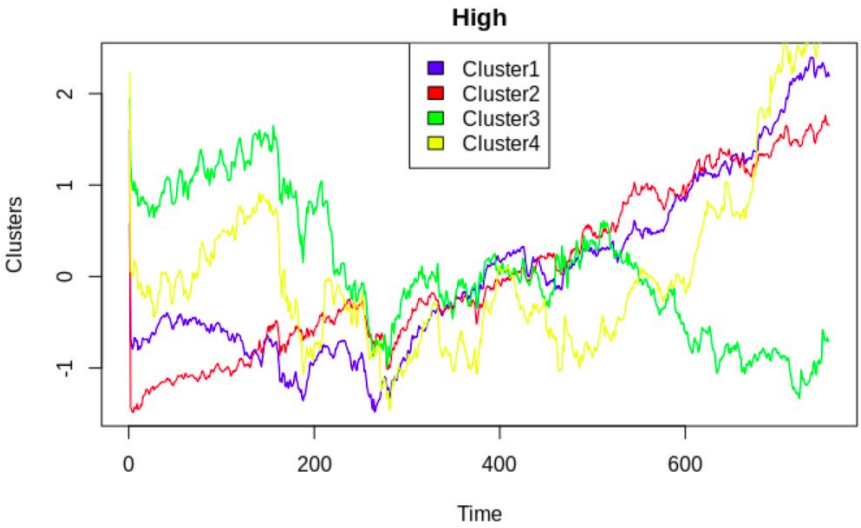
   - ❏ Data Set used : S&P 500 stock data
     - ❏ Data Set Characteristics : Multivariate, Time-Series
     - ❏ Number of Time Series: 20

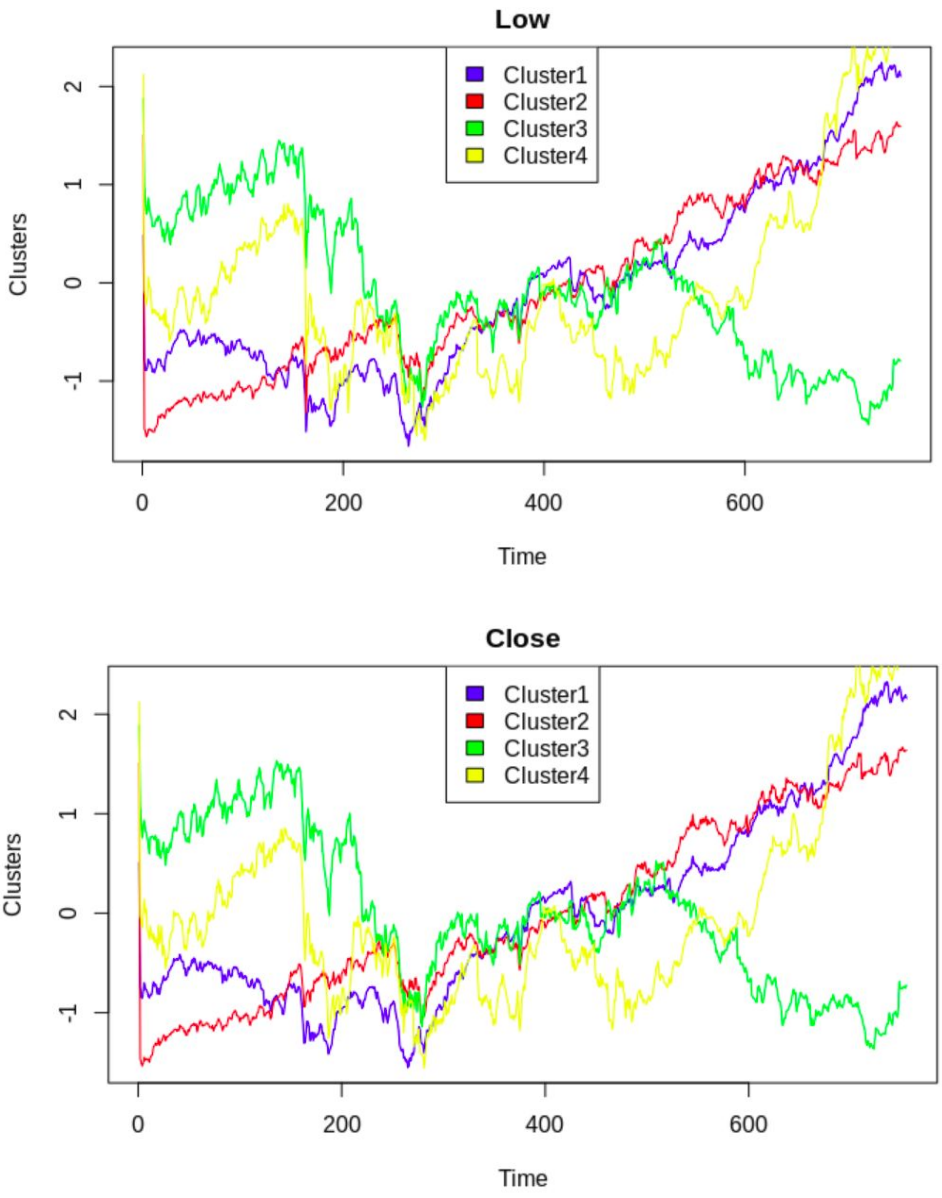
- ❑ Number of Instances in a Time Series: 755
- ❑ Number of Attributes : 4
- ❑ Attribute Characteristics : real
- ❑ Associated Tasks : clustering

The dataset contains files of data for individual stocks, labelled by their stock ticker name. All the files have the following columns :

- Date - in format: yy-mm-dd
- Open - Price of the stock at market open (this is NYSE data so all in USD)
- High - Highest price reached in the day
- Low - Lowest price reached in the day
- Close - Price of the stock at market close
- Name - the stock's ticker name







Figures showing changes in centroids w.r.t. Increasing iterations for k-means clustering for each of the attributes of the dataset (high, low, close and open)

Within Clusters Sum of Squares	20.9
Between Clusters Sum of Squares	86.1

Figure given below depicts final results of clustering with each company assigned to a particular cluster

stock_name <fctr>	clusters <int>	Name <fctr>	Sector <fctr>
MMM	1	3M Company	Industrials
AOS	2	A.O. Smith Corp	Industrials
ABT	4	Abbott Laboratories	Health Care
ABBV	4	AbbVie Inc.	Health Care
ABMD	2	ABIOMED Inc	Health Care
ACN	2	Accenture plc	Information Technology
ATVI	2	Activision Blizzard	Communication Services
ADBE	2	Adobe Inc.	Information Technology
AAP	3	Advance Auto Parts	Consumer Discretionary
AMD	2	Advanced Micro Devices Inc	Information Technology
AES	3	AES Corp	Utilities
AFL	1	AFLAC Inc	Financials
A	1	Agilent Technologies Inc	Health Care
APD	1	Air Products & Chemicals Inc	Materials
AKAM	3	Akamai Technologies Inc	Information Technology
ALK	2	Alaska Air Group Inc	Industrials
ALB	1	Albemarle Corp	Materials
ARE	1	Alexandria Real Estate Equities	Real Estate
ALXN	3	Alexion Pharmaceuticals	Health Care
ALGN	1	Align Technology	Health Care

20 rows

- **Observations** on the Clusters Formed
  - **IT** companies are contained in **Cluster 3**
  - **Materials** companies are contained in **Cluster 1**
  - **Industrial** companies are contained in **Cluster 2 or 1**
  - **Health Care** companies are **not clustered in any specific clusters**
  - Other categories do not not have required representation to interpret their clustering results

Conclusion

This report was an attempt to analyse multivariate time series data. We observed that a **Multivariate time series** has more than one **time-dependent variabl**e. Each variable depends not only on its past values but also has some dependency on other variables.

We first looked at **Vector Auto Regression** (VAR) - a multivariate forecasting algorithm that is used when two or more time series influence each other. It was used to implement **regression** on MVTS data. Using our dataset of Air Quality we predicted the future air quality index values based on the past data and observed that VAR is a robust model to predict future values for MVTS data.

We then shifted our focus to MVTS **classification** problem and observed that due to dependencies among variables, only the **k-NN** algorithm would work well. To measure similarity in the " Nearest-Neighbour " approach, we deployed the euclidean distance measure but realised that it is insufficient since it is prone to errors when encountering distortion in the time axis. Thus we applied **Dynamic Time Warping to euclidean distanc**e to improve the accuracy. To reduce the quadratic time complexity of DTW, two approaches were used- first. A **locality constrain**t was imposed upon the time series and second, **LB Keogh lower bound method** was used. This algorithm was applied on the EEG eye dataset using a single-nearest neighbour approach and binary classification results were obtained.

Our final task required us to **cluster** similar time series together. For this we used the already-existing **k-means** algorithm. The aim was to cluster similar companies in the S&P 500 index based on their historical stock prices dataset. As with classification, DTW euclidean distance was used to measure distance and sped up with the use of locality constraint and LB Keough lower bound method. The number of clusters was set beforehand and so was the initial choice of seeds. After a reasonable number of iterations, similar time series were clustered together.

So in this report we discussed various algorithms to analyse and modify MVTS data and implemented them in **R**. These implementations were used on real life applications to achieve satisfactory results.