**ASSIGNMENT TWO**

**Semester 2 - 2021**

**PAPER NAME: Data Mining and Machine Learning**

**PAPER CODE: COMP809**

**DUE DATE: Sunday 24 Oct 2021 at midnight**

**TOTAL MARKS: 100**

**Student Name: Bernard O'Leary**

**Student ID: 19075193**

**Note: This assignment must be complemented individually**

**Submission:** A soft copy needs to be submitted through Turnitin (a link for this purpose will be set up in Blackboard) Include your actual code (no screenshot) in Appendix with appropriate comments for each task.

**Instructions:**

1. **Academic Integrity Guidelines**

**The following actions may be deemed to constitute a breach of the General Academic Regulations Part 7: Academic Discipline,** **Section 2 Dishonesty During Assessment or Course of Study**

* 2.1.1 copies from, or inappropriately communicates with another person
* 2.1.3 plagiarises the work of another person without indicating that the work

is not the student’s own – using the full work or partial work of another person without giving due credit to the original creator of that work

* 2.1.4 Unauthorised collaboration in Assessment - collaborates with others in the preparation of material, except where this has been approved as an assessment requirement. This includes contract cheating where a student obtains services to produce or assist with an assessment
* 2.1.5 resubmits previously submitted work without prior approval of the exam board
* 2.1.6 Using any other unfair means

**Please email** [**DCT.EXAM@AUT.AC.NZ**](mailto:DCT.EXAM@AUT.AC.NZ) **if you have any technical issues with your online submission on Blackboard immediately**

Contents

[Part A 3](#_Toc86004251)

[Datasets and feature selection 3](#_Toc86004252)

[Sales Transaction Data 3](#_Toc86004253)

[Seoul Bike Data 3](#_Toc86004254)

[Water Treatment Data 4](#_Toc86004255)

[Task 1 4](#_Toc86004256)

[K means algorithm 4](#_Toc86004257)

[DBSCAN algorithm 6](#_Toc86004258)

[Agglomerative algorithm 7](#_Toc86004259)

[Task 2 8](#_Toc86004260)

[Answer A: which clustering algorithm performed best 8](#_Toc86004261)

[Answer B: why did it produced the best value for the CSM measure 9](#_Toc86004262)

[Answer C: which clustering algorithm is the overall winner 9](#_Toc86004263)

[Part B 9](#_Toc86004264)

[Pre-processing 9](#_Toc86004265)

[Feature selection (top 5) 9](#_Toc86004266)

[SO2 (�g/m�) 10](#_Toc86004267)

[NO2 (�g/m�) 10](#_Toc86004268)

[NO (�g/m�) 11](#_Toc86004269)

[Air Temp (�C) 11](#_Toc86004270)

[Summary statistics for PM2.5 (�g/m�) 12](#_Toc86004271)

[Table of summary stats for predictors 12](#_Toc86004272)

[Experimental methods 13](#_Toc86004273)

[Multilayer Perceptron (MLP) 13](#_Toc86004274)

[Answer 1: describe multilayer perceptron 13](#_Toc86004275)

[Answer 2: single hidden layer MLP with k= 25 neurons 14](#_Toc86004276)

[Answer 3: two hidden layer MLP with with k= 24 neurons (max) 15](#_Toc86004277)

[Answer 4: variation in the obtained performance metrics 16](#_Toc86004278)

[Long Short-Term Memory (LSTM) 16](#_Toc86004279)

[(1) 16](#_Toc86004280)

[(2) 16](#_Toc86004281)

[(3) 16](#_Toc86004282)

[(4) 16](#_Toc86004283)

[Model Comparison 16](#_Toc86004284)

[(1) 16](#_Toc86004285)

[(2) 16](#_Toc86004286)

[References 16](#_Toc86004287)

# Part A

The pre-processing performed was to bring all of the data in to Pandas dataframes, then apply categorisation to all the data so that it is all numeric and then go through the feature selection process. Following feature selection for each dataset, the selected features were nomalised prior to going through the algorithm.

For consistency, the PCA algorithm was used for feature selection for each of the datasets.

## Datasets and feature selection

### Sales Transaction Data

The Sales Transaction dataset contains weekly volumes of 800 products sold over a period of 52 week. Weeks are represented as columns and products are represented as rows. The dataset resembled a matrix, with a product column on the left. A normalised figure is provided for each record, however I have elected to normailise the entire dataset anyway.

The top three parameters were selected for this data as together they explained most of the variance.

Chart, histogram

Description automatically generated

### Seoul Bike Data

The Seoul Bike Sharing Demand dataset runs for between start of December 2017 and end of November 2019. There are over 8000 records, which describe the atmospheric conditions on the day and hour of the hirage, how many bikes were rented and whether the day was a holiday.

The top two parameters were selected for this data as together they explained most of the variance.

Chart, histogram

Description automatically generated

### Water Treatment Data

The Water Treatment dataset contains over 500 records and covers a range of parameters that describe the chemical composition of water treatment samples between March 1990 and August 1991 – 18 months worth of data.

The top five parameters were selected for this data as together they explained most of the variance.

Chart, histogram

Description automatically generated

## Task 1

### K means algorithm

#### Table A: K means algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Time taken (seconds) | CSM / SSE | K parameter |
| Sales Transactions Data | 0.39674830436706543 | 0.5529922804905653 / 14.437368116537462 | 2 |
| Water Treatment Data | 0.42758846282958984 | 0.1979742286699438 / 14.219162585579324 | 6 |
| Seoul Bike Data | 5.6746766567230225 | 0.2736422972611256 / 297.7696053888172 | 2 |

#### Sales Transactions Data CSM chart for KMeans

Chart, scatter chart

Description automatically generated

#### Water Treatment Data CSM chart for KMeans

Chart, scatter chart

Description automatically generated

#### Seoul Bike Data CSM chart for KMeans

Chart, histogram

Description automatically generated

### DBSCAN algorithm

#### Table B: DBSCAN algorithm

For the DBSCAN algorithm I needed to adjust the EPS value also, but unfortunately I was unable to get a correct value of the EPS so that the algorithm would complete correctly for the Water Treatment Data and Seoul Bikes Data.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Time taken (seconds) | CSM / SSE | EPS parameter |
| Sales Transactions Data | 0.346055269241333 | 0.35714256858302806 / 27.772162712889894 | 0.2 |
| Water Treatment Data | [Incomplete] | 0.01814249370553166 / [Incomplete] | 0.3 |
| Seoul Bike Data | [Incomplete] | -0.48965343976481296 / [Incomplete] | 0.1 |

#### Sales Transactions Data CSM chart for DBSCAN

Chart, scatter chart

Description automatically generated

### Agglomerative algorithm

#### Table C: Agglomerative algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Time taken (seconds) | CSM / SSE | Clusters parameter |
| Sales Transactions  Data | 0.23120903968811035 | 0.5630813204653953  / 14.907766709758254 | 2 |
| Water Treatment Data | 0.12308955192565918 | 0.15524541375205828 / 24.894066528279865 | 2 |
| Seoul Bike Data | 8.33217167854309 | 0.24688690435967905  / 308.6434248773903 | 2 |

#### Sales Transactions Data CSM chart for Agglomerative

Chart, scatter chart

Description automatically generated

#### Water Treatment Data CSM chart for Agglomerative

Chart, scatter chart

Description automatically generated

#### Seoul Bike Data CSM chart for Agglomerative

Chart, histogram

Description automatically generated

## Task 2

### Answer A: which clustering algorithm performed best

The KMeans algorithm appears to perform slower than the aglomorative algorithm in general, although not in all cases. Unfortunately I struggled with the DBSCAN algorithm so have incomplete data for this experiment. For that reason I am only comparing the Aglomorative algorithm and K Means algorithms.

For the Sales Transactions Data the Aglomorative algorithm performed best, with CSM of 0.56 and time taken of 0.23 seconds, for 2 clusters.

For the Water Treatment Data the K Means algorithm performed best, with CSM of 0.197 and time taken of 0.43 seconds, for 2 clusters.

For the Seoul Bike Data the K Means algorithm performed best, with CSM of 0.27 and time taken of 5.6 seconds, for 2 clusters.

I am using a combination of time taken and CSM score to rank the algorithms. The higher the CSM score, the better the algorithm. If the CSM score is high and the time taken is low, then that algorithm is considered the better one.

### Answer B: why did it produced the best value for the CSM measure

Looking at the shape of the data, DBSCAN might have been good to apply to the Seoul Bike dataset as it appears to have some overlap in it’s structure. Unfortunately I have have not been able to test this on the dataset. The best performance for the Seoul Bike dataset was K Means.

The result I had for the Water Treatment data was the K Means was the better algorithm. This was somewhat surprising as looking at the dataset, it seems relatively even distributed with no noticeable clustering. I would have thought that the Aglomorative algorithm would have performed better for this dataset.

The result I had for the Sales Transaction data indicated that the Agglomorative algorithm slightly outperformed the K Means algorithm. This dataset is quite neatly dividied in to two clusters, which probably makes it equally good for a hierarchical, a divisive or a partitional algorithm.

### Answer C: which clustering algorithm is the overall winner

Of the three algorithms, K Means seemed easiest to work with and to understand. K Means also generally performed the quickest of the three algorithms although I did not get to test DBSCAN to a significant extent.

# Part B

## Pre-processing

I have used the file Penrose\_Hourly\_AggregateData\_Jan2016Dec2020.csv that was provided for the assignment.

To be able to include the hourly segmentation, I needed to do feature engineering to produce four new columns from the Timestamp column. This was done y concerting the Timestamp type to a datetime and then getting the hour, day, month and year values form the datetime and creating new columns.

I also followed the guidance in the assignment and removed values of greater than 100 from the relatively humidity (%) and temperatures of above 40 degrees Celsius from the Air Temp column.

Finally, all columns were converted to categorical. Initially I wasn’t going to do this, but the MLP compained that the target categories were floats. I ended up “binning” these values anyway to reduce the number of possible categories, so this step may not have been necessary.

## Feature selection (top 5)

The following sections provide details of the five features that were selected for the predictor variables for the experiment due to their being the variables that had the highest correlation in comparison to the target variable when assessed indicidually against the target (PM2.5 (�g/m�)) using linear regression. A corellation plot for the linear regression model for each predictor is provided.

The statistics for each predictor are provided, including the Pearson Correlation which describes each individual attribute’s influence on the target variable.

The reason I used this approach is because it seemed like the most intuitive and simplistic way to get a very clear view of what each predictor variable looked like when it was plotted against the target. An alternative approach would have been to plot all of the predictor vaiables against the target in a single table.

### SO2 (�g/m�)

Slope: 1.011

Intercept: 106.215

explained\_variance: 0.1236

r2: 0.1236

MAE: 36.4505

MSE: 2458.9511

RMSE: 49.5878

Pearsons correlation: 0.3515

Chart, scatter chart

Description automatically generated

### NO2 (�g/m�)

Slope: 0.128

Intercept: 98.190

explained\_variance: 0.1137

r2: 0.1137

MAE: 36.438

MSE: 2486.5994

RMSE: 49.8658

Pearsons correlation: 0.3372

Chart, scatter chart

Description automatically generated

### NO (�g/m�)

Slope: 0.113

Intercept: 103.757

explained\_variance: 0.1904

r2: 0.1904

MAE: 35.2568

MSE: 2271.5162

RMSE: 47.6604

Pearsons correlation: 0.4363

Chart, scatter chart

Description automatically generated

Wind Direction (�)

Intercept: 106.028

explained\_variance: 0.0211

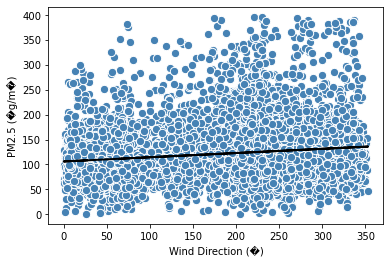
r2: 0.0211

MAE: 38.016

MSE: 2746.4533

RMSE: 52.4066

Pearsons correlation: 0.1452



### Air Temp (�C)

Slope: -3.581

Intercept: 172.655

explained\_variance: 0.0722

r2: 0.0722

MAE: 37.9256

MSE: 2603.0704

RMSE: 51.0203

Pearsons correlation: -0.2687

Chart, scatter chart

Description automatically generated

### Summary statistics for PM2.5 (�g/m�)

The following summary statistics were attained for the target variable by using the .describe() function against the variable, as is provided by the Pandas package.

count 10313.000000

mean 121.846310

std 52.970365

min 0.000000

25% 88.000000

50% 113.000000

75% 145.000000

max 397.000000

Name: PM2.5 (�g/m�), dtype: float64

### Table of summary stats for predictors

The formula for Pearson Corrleation is as follows:

A picture containing table

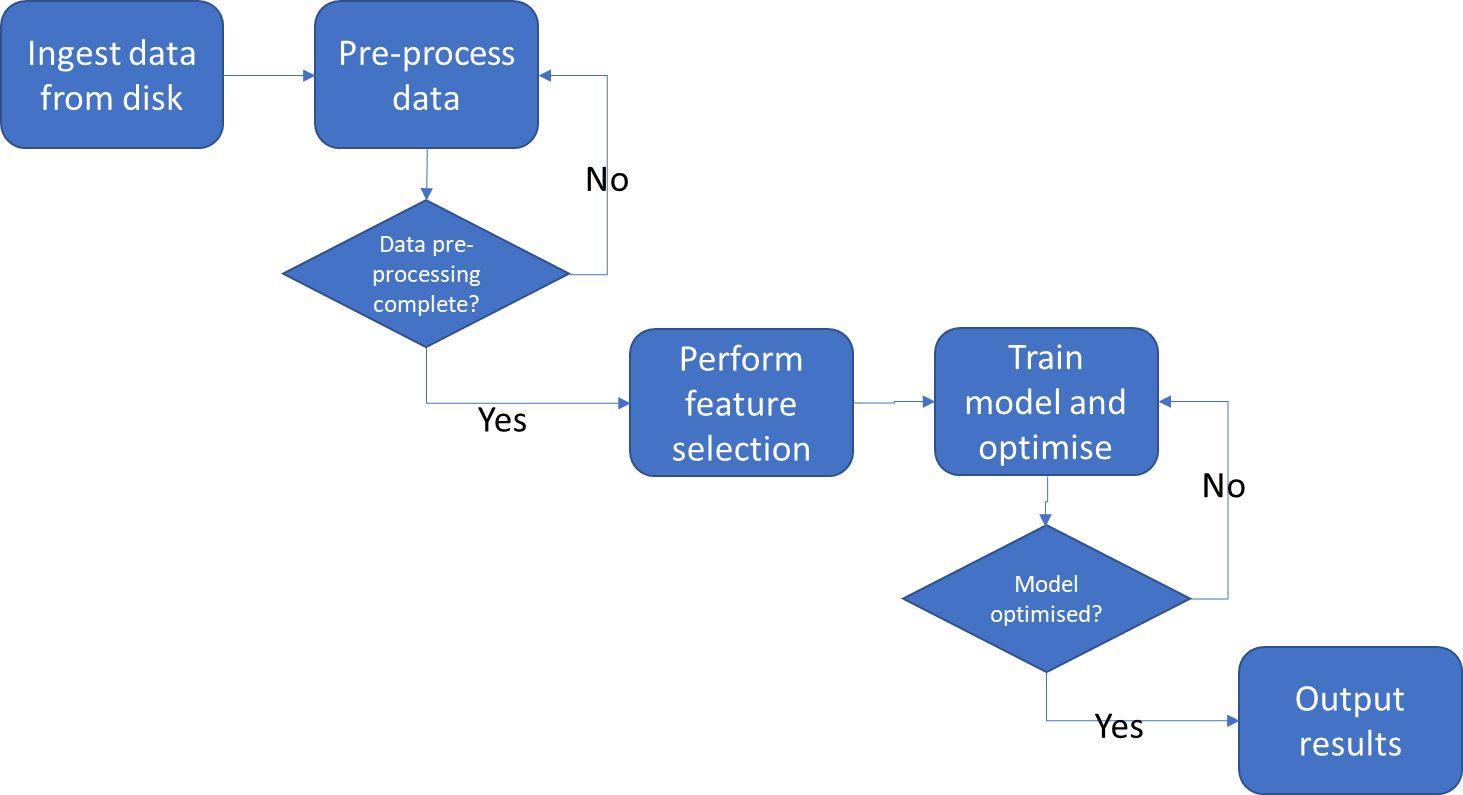
Description automatically generated

The five best matched variables based on Pearson Correlation are covered in the following table. The Pearson Correlation value provided is the result of a linear regression for each variable against the target. Higher values are better and generally values that are higher in each direction than 0.5 or -0.5 are considered highly correlated.Unfortunately I didn’t get any results that were highly correlated.

|  |  |
| --- | --- |
| Predictor variable | Pearson Correlation value |
| SO2 (�g/m�) | 0.3515 |
| NO2 (�g/m�) | 0.3372 |
| NO (�g/m�) | 0.4363 |
| Wind Direction (�) | 0.1452 |
| Air Temp (�C) | -0.2687 |

## Experimental methods

The following diagram issulstrates the process used to generated the model that is created for the MLPClassifier and the LTSM model. This is a high-level overview of the process, and detailed steps such as the train/test-split are incorporated into the bigger parts of the process.

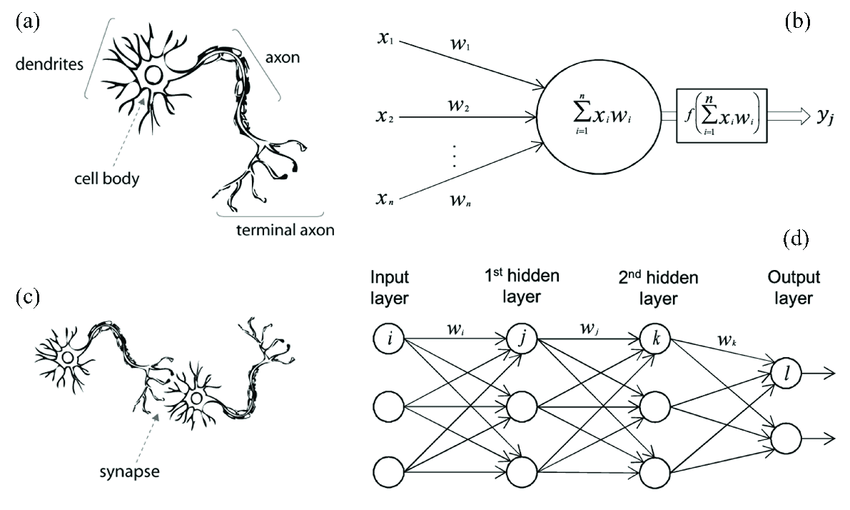


## Multilayer Perceptron (MLP)

## Answer 1: describe multilayer perceptron

The Multilayer Perceptron is a machine learning approach that tries to mimic the process biological process that is used by brain to achieve predictive capability based on incoming information.

The perceptron element of the model mimics the behaviour of a biological neuron in a brain, and synapses, which link neorons together.



This image is taken from Meng Z., et.al. [1] provides an overview for the model of the synapse and the neuron that the Multi Layer Perceptron borrows from.

The neuron is modelled by the “perceptron” by emulating input signals, applying a weight to each signal, summing the values of the output of the signals multiplied by the respective weights and then deciding of it will in turn output a signal based on whether the sum of the values input and calculated cross a threshold, defined by another function such as the Sigmod function. The neuron in the brain performs this work using input chemical signal, whereas the perceptron performs work work using digital signals. Perceptrons are combined in layers to mimic the function of synapses, which link neurons in the brain.

Artificial neural networks have been around as a concept for a relatively long time, but have become increasingly popular and viable in recent years with the availability of highly parallel computing models such as GPU processors and cloud computing. Because an artificial neural networks mimic the biological brain which is a massively parallel computing process, the ANN also requires platforms that support highly parallel worksloads.

A limitation of the ANN is that it is not inherently explainable in that the process that the training algoritm applies to produce the optimised model is very difficult to interpret in terms of understanding which aspect of the model is responsible for a particular outcome.

### Answer 2: single hidden layer MLP with k= 25 neurons

The single layer network with 25 nerons in the single layer took several minutes to run and yielded a model with accuracy of approximately 0.45 across 7 target categories. The target categories were reached by “binning” the target variable, which was distributed across many values between 0 and 397, resulting initially in a very low accuracy model. I therefore decided to apply binning and split the variable in to 7 evenly split categories with range of 50 values, between 0 and 400. This was achieve as follows:

y = np.digitize(y,bins=[50,100,150,200,250,300,350])

Without bining, the accuracy of the model dropped to 0.01, with binning across 4 categories, the accuracy of the model increases to about 0.6. & seemed like an acceptable compromise.

Details of the model are provided below, along with charts and metrics.

Graphical user interface, text, application

Description automatically generated

Loss 1.4461270468227754

iterations 78

Assigned classes [0 1 2 3 4 5 6 7]

Shape, square

Description automatically generated

Accuracy score: 0.44

### Answer 3: two hidden layer MLP with with k= 24 neurons (max)

The following table provides a summary of the result of running the algorithm across 25 iterations of the process of incrementing the first layer of a 2 layer MLP by one and decrementing the second layer by one, with a starting value of 1 for the first layer and 24 for the second layer, until the values are swapped.

|  |  |
| --- | --- |
| Combination of neurons | Accuracy |
| 17,8 | 0.450226 |
| 6,19 | 0.446025 |
| 16,9 | 0.445055 |
| 11,14 | 0.443439 |
| 19,6 | 0.441823 |
| 9,16 | 0.4415 |
| 10,15 | 0.441176 |
| 1,24 | 0.43956 |
| 7,18 | 0.43956 |
| 14,11 | 0.43245 |
| 13,12 | 0.430834 |
| 8,17 | 0.425339 |
| 22,3 | 0.423077 |
| 18,7 | 0.420491 |
| 21,4 | 0.419522 |
| 23,2 | 0.419198 |
| 24,1 | 0.419198 |
| 3,22 | 0.418875 |
| 20,5 | 0.417582 |
| 12,13 | 0.413704 |
| 15,10 | 0.411441 |
| 4,21 | 0.410472 |
| 2,23 | 0.389787 |
| 5,20 | 0.378474 |

### Answer 4: variation in the obtained performance metrics

As can be seen from the table provided in the (3) part of this section, although there is variation in the results, no obvious pattern has emerged – i.e. there is no marked difference based on the number of perceptrons in the hidden layers being about the same or being very different, the result seems quite random. The two lowest results where where the first hidden layer had low number of perceptrons, but also one of the highest results was where the first layer has 1 perceptron (and the second layer had 24).

It is difficult to say which architecture gives the better performance, or to explain the results, as no pattern has emerged from this experiment and therefore I am unable to make any assumptions about what might (or might not) be happening. What I can say though is that since the pattern is randm, perhaps for this dataset a deeper network is onconsequential.

## Long Short-Term Memory (LSTM)

asdf

### (1)

DNC

### (2)

DNC

### (3)

DNC

### (4)

DNC

## Model Comparison

DNC

### (1)

DNC

### (2)

DNC

# References

[1] Meng Z, Hu Y, Ancey C. “Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows.” Water. 2020; 12(2):600. <https://doi.org/10.3390/w12020600>

# Appendix

**Four appendixes are provided**

19075153\_O'Leary\_PartA1\_AM: extract from Jupyter-Labs for answers to Part A (Agglomerative)

19075153\_O'Leary\_PartA1\_DB: extract from Jupyter-Labs for answers to Part A (DBSCAN)

19075153\_O'Leary\_PartA1\_KM: extract from Jupyter-Labs for answers to Part A (Kmeans)

19075153\_O'Leary\_PartB: extract from Jupyter-Labs for answers to Part B