BITI3133: NEURAL NETWORK

SEM 2, 2023/2024

MINI PROJECTS (30%)

PREDICTIVE MAINTENANCE FOR HVAC SYSTEMS IN A COMMERCIAL BUILDING

LECTURER NAME: DR. BURHANUDDIN MOHD ABOOBAIDER

NAME	MATRIC NO
Saedatul Izzah bt Azizul Rahman	B032110312
‘Iffah Nafisah Binti Azizan	B032110297
Marrylind Grace Ann Anak John	B032110502
Muhammad Arif Zaqwan Bin Mohd Zaki	B032110337

Table of Contents

Introduction

Predictive maintenance plays a crucial role in ensuring the smooth operation and optimal performance of HVAC systems in commercial buildings. HVAC systems are responsible for maintaining a comfortable indoor environment by regulating temperature, humidity, and air quality. However, like any complex machinery, these systems are susceptible to faults, malfunctions, and inefficiencies that can disrupt operations and increase energy consumption.

This case study focuses on the development and evaluation of a classifier model for predictive maintenance in HVAC systems in a commercial building. By harnessing the power of advanced analytical models, such as machine learning algorithms and neural networks, we aim to develop a proactive maintenance strategy that improves equipment reliability, minimizes downtime, optimizes energy consumption, and enhances occupant comfort.

The primary objective of this case study is to train a classifier model that can analyze data collected from the HVAC system and predict maintenance needs. By analyzing key variables such as temperature, humidity, power consumption, and energy usage, the model will identify patterns, anomalies, and potential faults in the system. This will allow maintenance teams to take proactive measures, such as scheduling maintenance activities and addressing issues before they escalate into major failures.

In addition to improving maintenance practices, this case study also aims to optimize energy efficiency in HVAC systems. By detecting inefficiencies and deviations from optimal performance, the predictive maintenance model can provide insights to optimize energy consumption, reduce waste, and achieve cost savings.

Throughout this case study, we will delve into the details of the dataset used, the analytical techniques employed, and the decision-making processes involved. We will explore how advanced analytical models, including machine learning algorithms and neural networks, can be utilized to analyze the dataset, train predictive maintenance models, and make informed decisions based on the model's predictions.

Detail description about sample data

how to download

Sample Data Description: The dataset utilized in this case study focuses on predictive maintenance of HVAC systems in a commercial building. It comprises 10 variables, each with its own unique significance.The dataset was retrieved from Mendeley Data (refer Appendix A),:

1. Return Air Temperature(°C) :This variable represents the temperature of the air being returned to the HVAC system. It helps in covering the thermal conditions within the structure.

2. Supply Air Temperature(°C) This variable represents the temperature of the air being supplied by the HVAC system into the structure space. It indicates the effectiveness of the cooling or heating process

3. Outdoor Air Temperature(°C) This variable represents the temperature of the out-of-door air. It's a pivotal factor in determining the effectiveness of the HVAC system, especially for systems that incorporate out-of-door air for ventilation or cooling purposes.

4. Return Air Humidity() This variable represents the relative moisture of the air being returned to the HVAC system. It provides perceptivity into the humidity content within the structure
.

5. Supply Air Humidity() This variable represents the relative moisture of the air being supplied by the HVAC system. It indicates the position of moisture control achieved by the system.

6. Outdoor Air Humidity() This variable represents the relative moisture of the out-of-door air. It helps in assessing the out-of-door air's impact on the HVAC system's performance and the inner terrain.

7. Return Air Temperature Setpoint(°C) This variable represents the asked temperature setpoint for the return air. It serves as a reference for comparing the factual return air temperature and assessing the system's performance.

8. Humidifier Saturation Temperature(°C) This variable represents the achromatism temperature in the humidifier, which is used to control the moisture position in the structure. It provides perceptivity into the humidification process.

9. Fan Power(kW) (Power)This variable represents the power needed by the suckers in the HVAC system. It indicates the energy consumption associated with the addict operation.

10. Fan Energy(kWh)(Energy) :This variable represents the energy needed by the suckers in the HVAC system over a specific period. It provides an overall measure of addict energy consumption.

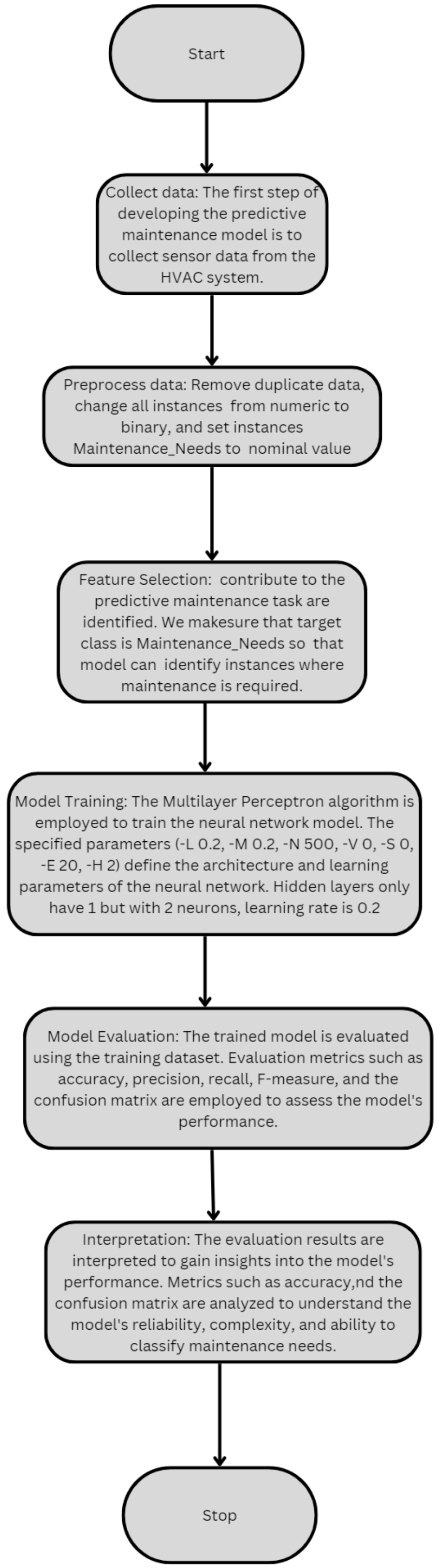
11. Maintanance_Needs: The "Maintenance_Needs" variable represents whether maintenance is needed (1) or not needed (0)

The sample data provides a comprehensive set of variables that capture important aspects of HVAC system operation, including temperature, moisture,and setpoints,. These variables enable analysis and modeling to identify patterns, anomalies, and implicit faults within the system. Using this sample data, we can gain insight into the HVAC system's performance, identify deviations from normal operating conditions, and develop predictive conservation models that can predict failures and optimize conservation schedules.Originally we retrieved the data from the website (refer Appendix A).

However we dont really utilize the exact data from the provided dataset instead we create a dummy dataset but fulfill our case study requirement that now consisting 132 instances.
This is step to download our created dummy dataset :

1.) Together with the report, there is attachment called “HVAC (Dummy Data)” where it store all raw data that is yet to be preprocessed inside Weka tools. This data converted into csv commas where Weka can read and do the preprocesing
2.) Click the attachment and ,can view all the new instances but with same attributes as original data from the website.

Flow chart and learning process



Step by step learning and analysis

1. Collect data: Collect data from sensors and automation systems of HVAC system.
2. Data preprocessing: Clean, preprocess, and transform the collected raw data into structured form..
3. Select feature variables: Define the input parameters and identify capable feature variables to improve the anomaly detector's performance.
4. Model Configuration: Configure the MLP algorithm in Weka by specifying the relevant options and parameters. This includes setting the number of hidden layers, the number of neurons in each layer, the learning rate, the momentum, and other hyperparameters.
5. Training the Model: Apply the MLP algorithm to the prepared dataset to train the neural network model. Weka provides a user-friendly interface or command-line options to initiate the training process. During training, the model iteratively adjusts the weights based on the provided learning rate and the observed errors, aiming to minimize the overall prediction error.
6. Interpretation and Analysis: Analyze the trained MLP model to gain insights into the relationships between the input variables and the predicted output

How analysis can be conducted

The dataset utilized in this study comprises various attributes related to temperatures (return, supply, and outdoor air), relative humidities, temperature setpoints, and power/energy consumption of fans. By training the MLP model on this dataset, we can discover the inherent connections between these attributes and maintenance requirements.

In this study, we utilized the Weka tool to perform the analysis. Weka is a popular and powerful open-source data mining and machine learning software that provides a user-friendly interface for data analysis tasks.

With Weka, we were able to preprocess the dataset by handling missing values, outliers, and inconsistencies. We also conducted data exploration to gain insights into the distribution, correlations, and patterns in the data.

We then selected the Multilayer Perceptron (MLP) algorithm from Weka's wide range of available machine learning algorithms. The MLP algorithm is well-suited for building neural network models and has been widely used in various domains.

Using Weka, we applied learning rates to the MLP model and analyzed the resulting evaluation metrics such as accuracy, precision, recall, F1 score, and ROC curves

During the analysis, we assessed the MLP model's performance by employing learning rates. The findings demonstrated a strong correlation between predicted and actual maintenance needs.

Findings and decision

Based on the output of the evaluation on the training set, here are the findings and decisions that can be made:How to run this output shown on Appendix B.

1. Model Performance:

```
=== Summary ===

Correctly Classified Instances        108               81.8182 %
Incorrectly Classified Instances       24               18.1818 %
Kappa statistic                         0.6296
```

- The Multilayer Perceptron classifier achieved an overall accuracy of 81.82% on the training set.
- The model correctly classified 108 instances (81.82%) and misclassified 24 instances (18.18%).
- The Kappa statistic, which measures the agreement between the predicted and actual classes, is 0.6296. This indicates a moderate level of agreement.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.804	0.171	0.776	0.804	0.789	0.630	0.805	0.705	0
	0.829	0.196	0.851	0.829	0.840	0.630	0.805	0.803	1
Weighted Avg.	0.818	0.186	0.819	0.818	0.819	0.630	0.805	0.761	

2. Class-Specific Performance:

- Class 0 (No Maintenance Needs):

- True Positive Rate (Recall): 80.4%

- False Positive Rate: 17.1%

- Precision: 77.6%

- F-Measure: 78.9%

- Class 1 (Maintenance Needs):

- True Positive Rate (Recall): 82.9%

- False Positive Rate: 19.6%

- Precision: 85.1%

- F-Measure: 84.0%

- The model shows reasonably good performance in classifying both classes, with slightly better results for class 1 (Maintenance Needs).

```
Mean absolute error                      0.3225
Root mean squared error                  0.4054
Relative absolute error                 65.9964 %
Root relative squared error             82.0343 %
Total Number of Instances              132
```

3. Evaluation Metrics:

- Mean Absolute Error (MAE): The average absolute difference between the predicted and actual values is 0.3225. This represents the average magnitude of the model's errors.
- Root Mean Squared Error (RMSE): The square root of the average squared difference between the predicted and actual values is 0.4054. It provides a measure of the overall model's prediction error.
- Relative Absolute Error: The MAE as a percentage of the mean of the actual values is 65.9964%. It indicates the average prediction error relative to the range of the target variable.
- Root Relative Squared Error: The RMSE as a percentage of the range of the actual values is 82.0343%. It provides a measure of the average prediction error relative to the range of the target variable.

```
=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    1.8085119670930399
    Node 2    -4.038269779799336
    Node 3    -3.6723335013571807
Sigmoid Node 1
    Inputs    Weights
    Threshold    -1.8085119681414261
    Node 2    4.03826978234931
    Node 3    3.6723335035136224
Sigmoid Node 2
    Inputs    Weights
    Threshold    -6.3495511706213
    Attrib T_Supply    -4.9240479347289
    Attrib T_Return    0.3284469259310074
    Attrib SP_Return    0.5247463691168623
    Attrib T_Saturation    -1.1176809002354324
    Attrib T_Outdoor    -9.309029394054594
    Attrib RH_Supply    1.2118539275413578
    Attrib RH_Return    -1.6334588152671397
    Attrib RH_Outdoor    -10.233062067286314
    Attrib Energy    6.638797180394104
    Attrib Power    4.908285899080868
Sigmoid Node 3
    Inputs    Weights
    Threshold    -1.7485803071406811
    Attrib T_Supply    9.984825506873612
    Attrib T_Return    -2.8955410641204162
```

```
Sigmoid Node 3
    Inputs    Weights
    Threshold    -1.7485803071406811
    Attrib T_Supply    9.984825506873612
    Attrib T_Return    -2.8955410641204162
    Attrib SP_Return    -0.49946605207049444
    Attrib T_Saturation    -5.122981961679337
    Attrib T_Outdoor    -3.1313073784616656
    Attrib RH_Supply    5.718815983047056
    Attrib RH_Return    0.49302820425997446
    Attrib RH_Outdoor    2.5169990477148407
    Attrib Energy    0.9953788172031591
    Attrib Power    -8.316234991489974
Class 0
    Input
    Node 0
Class 1
    Input
    Node 1
```

4. Model Interpretation:

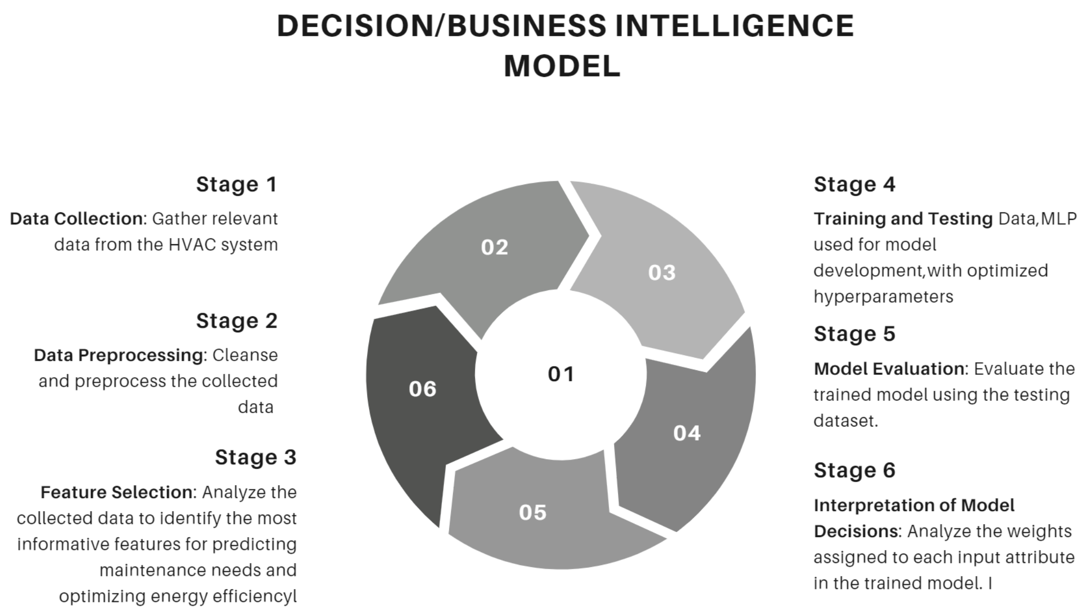
- The trained Multilayer Perceptron model consists of four sigmoid nodes and two output classes.
- Each sigmoid node has inputs, weights, and a threshold that determines its activation.
- The model's architecture and weights indicate the relationships between the input attributes and the output classes.
- Sigmoid nodes 2 and 3 have the most significant influence on the classification decisions.

5. Decision:

- Based on the evaluation results, the Multilayer Perceptron model shows promising performance in classifying maintenance needs in HVAC systems.
- The model achieves a reasonably high accuracy and exhibits a good balance between precision and recall for both classes.

Decision/business intelligence model

Business or Intelligence Model:



Conclusion

Based on our evaluation of the Multilayer Perceptron classifier, we found that our predictive maintenance model for HVAC systems shows promising performance. With an overall accuracy of 81.82%, the model effectively classifies instances into "No Maintenance Needs" and "Maintenance Needs" classes. The recall and precision rates indicate its ability to accurately identify instances in each class.

The evaluation metrics, such as MAE, RMSE, Relative Absolute Error, and Root Relative Squared Error, provide insights into the model's prediction errors and its performance relative to the target variable. These metrics help us understand the model's overall effectiveness.

Integrating this model into our decision-making process enables us to enhance maintenance practices, optimize energy efficiency, and improve system performance. Predictive maintenance allows us to proactively identify and address faults and inefficiencies, preventing major failures and ensuring smooth operation.

Implementing this proactive maintenance strategy in our HVAC systems offers numerous benefits, including increased reliability, minimized downtime, improved energy consumption, and enhanced occupant comfort. By leveraging the model's predictions and insights, we can make informed decisions on maintenance activities, energy optimization strategies, and overall system management.

Overall, our predictive maintenance model based on the Multilayer Perceptron algorithm provides a solid foundation for optimizing maintenance and energy efficiency in our HVAC systems. It contributes to the cost-effectiveness and smooth operation of our commercial building, allowing us to prioritize resources effectively and ensure optimal performance.

References

Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L., & Bennadji, B. (2021). Predictive Maintenance in Building Facilities: A Machine Learning-Based Approach. *Sensors*, *21*(4), 1044. https://doi.org/10.3390/s21041044

Borda, D., Bergagio, M., Amerio, M., Masoero, M. C., Borchiellini, R., & Papurello, D. (2023). Development of Anomaly Detectors for HVAC Systems Using Machine Learning. *Processes*, *11*(2), 535. https://doi.org/10.3390/pr11020535

HowTo. (2017, July 28). *Multilayer Perceptron | Neural Network in Weka : Weka Tutorials # 5* [Video]. YouTube. https://www.youtube.com/watch?v=Gw5s3yYVQAE

Gonfalonieri, A. (2021, December 12). How to Implement Machine Learning For Predictive Maintenance. *Medium*. https://towardsdatascience.com/how-to-implement-machine-learning-for-predictive-maintenance-4633cdbe4860

Appendix A

The dataset was retrieved from :

Borda, D. (2022). Development of Anomaly Detectors for HVAC Systems using Machine Learning. *Mendeley Data*. https://doi.org/10.17632/mjhr46dkj6.1

Sample of raw data:

	A	B	C	D	E	F	G	H	I	J
1	T_Supply	T_Return	SP_Returr	T_Saturati	T_Outdoo	RH_Suppl	RH_Returr	RH_Outdc	Energy	Power
2	19.2	22.26	21.5	17.32	24.3	34.44	23.39	30	13	4.932
3	19.86	22.31	21.5	17.58	24.3	34.11	22.96	31	12	4.932
4	19.71	22.57	21.5	17.52	24.3	33.9	22.78	30	12	4.956
5	20.075	22.64	21.5	17.84	24.3	33.9	22.7	29	13	4.968
6	20.085	22.7	21.5	18.68	24.3	33.4	22.24	29	14	5.064
7	19.96	22.93	21.5	18.58	23.3	33.09	22.05	29	12	5.04
8	20.225	23.075	21.5	18.72	23.3	32.91	21.91	29	12	4.944
9	20.24	23.095	21.5	18.7	24.3	33.11	21.79	29	13	5.04
10	20.17	23.13	21.5	17.92	23.3	32.63	21.64	30	12	5.016
11	20.38	23.14	21.5	18.38	22.1	32.81	21.32	31	12	4.992
12	20.575	23.12	21.5	18.32	23.1	32.7	21.6	29	13	4.98
13	21.055	23.11	21.5	18.8	22.1	32.37	21.6	27	13	5.064
14	20.595	23.165	21.5	18.64	23.1	32.27	21.47	29	12	4.992
15	20.665	23.11	21.5	18.82	24.1	32.96	21.63	31	12	4.992
16	20.525	22.97	21.5	18.46	24.1	32.85	21.85	32	13	5.04
17	20.39	22.84	21.5	18.44	24.1	32.92	21.89	33	12	4.98
18	20.645	22.93	21.5	18.46	23.1	32.37	21.62	37	12	5.004
19	21.34	22.7	21.5	18.92	21.1	32.2	21.8	41	14	4.944
20	19.93	22.81	21.5	18.08	22.1	33.86	21.76	41	13	5.028

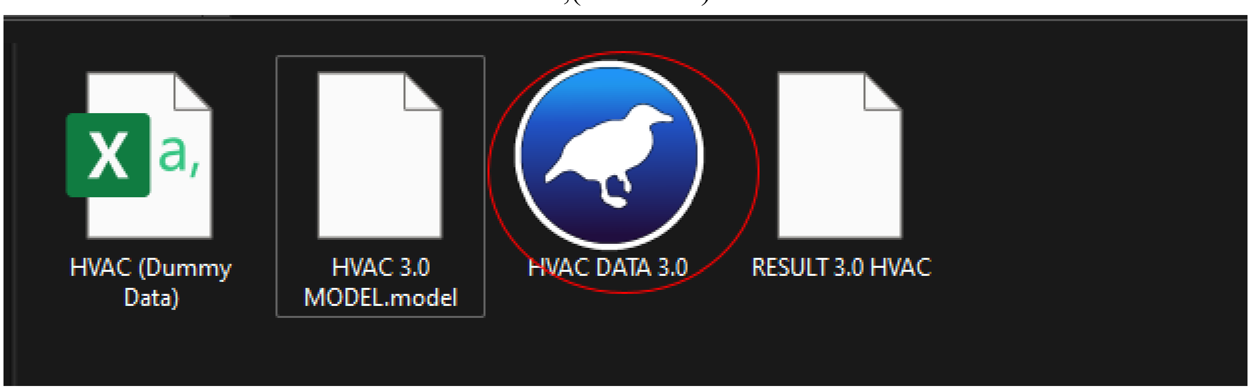
Sample of few our used data:

	A	B	C	D	E	F	G	H	I	J	K	L
1	T_Supply	T_Return	T_Supply	T_Supply	T_Outdoo	RH_Suppl	RH_Returi	RH_Outdc	Energy	Power	Maintanance_Needs	
2	19.2	22.26	21.5	17.32	24.3	34.44	23.39	30	13	4.932	1	
3	19.86	22.31	21.5	17.58	24.3	34.11	22.96	31	12	4.932	0	
4	19.71	22.57	21.5	17.52	24.3	33.9	22.78	30	12	4.956	1	
5	20.075	22.64	21.5	17.84	24.3	33.9	22.7	29	13	4.968	0	
6	20.085	22.7	21.5	18.68	24.3	33.4	22.24	29	14	5.064	1	
7	19.96	22.93	21.5	18.58	23.3	33.09	22.05	29	12	5.04	1	
8	20.225	23.075	21.5	18.72	23.3	32.91	21.91	29	12	4.944	0	
9	20.24	23.095	21.5	18.7	24.3	33.11	21.79	29	13	5.04	1	
10	20.17	23.13	21.5	17.92	23.3	32.63	21.64	30	12	5.016	0	
11	20.38	23.14	21.5	18.38	22.1	32.81	21.32	31	12	4.992	1	
12	20.575	23.12	21.5	18.32	23.1	32.7	21.6	29	13	4.98	0	
13	21.055	23.11	21.5	18.8	22.1	32.37	21.6	27	13	5.064	1	
14	20.595	23.165	21.5	18.64	23.1	32.27	21.47	29	12	4.992	1	
15	20.665	23.11	21.5	18.82	24.1	32.96	21.63	31	12	4.992	0	
16	20.525	22.97	21.5	18.46	24.1	32.85	21.85	32	13	5.04	1	
17	20.39	22.84	21.5	18.44	24.1	32.92	21.89	33	12	4.98	0	
18	20.645	22.93	21.5	18.46	23.1	32.37	21.62	37	12	5.004	1	
19	21.34	22.7	21.5	18.92	21.1	32.2	21.8	41	14	4.944	1	
20	19.93	22.81	21.5	18.08	22.1	33.86	21.76	41	13	5.028	0	
21	20.275	22.7	21.5	18.5	22.1	33.55	21.92	41	12	5.004	1	

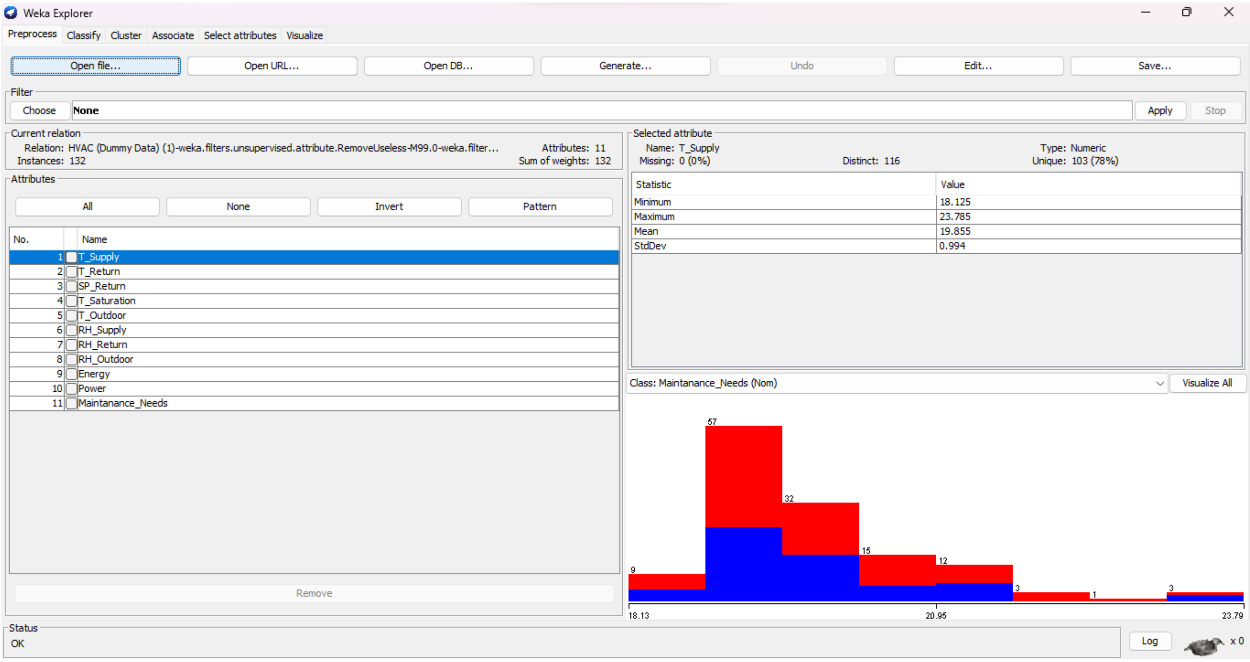
User Manual :

Make sure to have Weka tools inside desktop to fully run this.

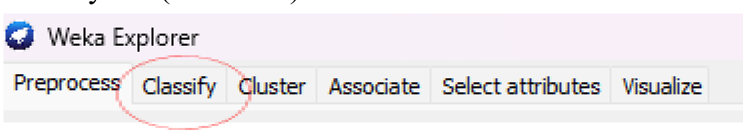
1. Click file name “HVAC DATA 3.0”,(red circle):



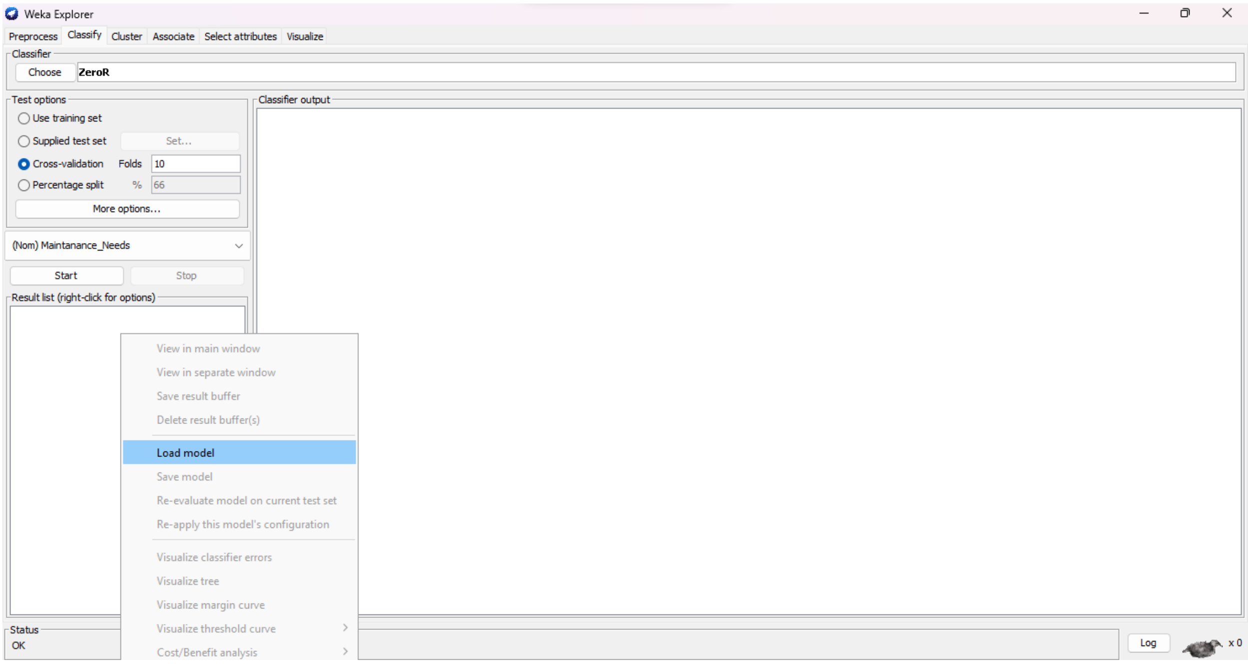
2. It will open at Weka like this:

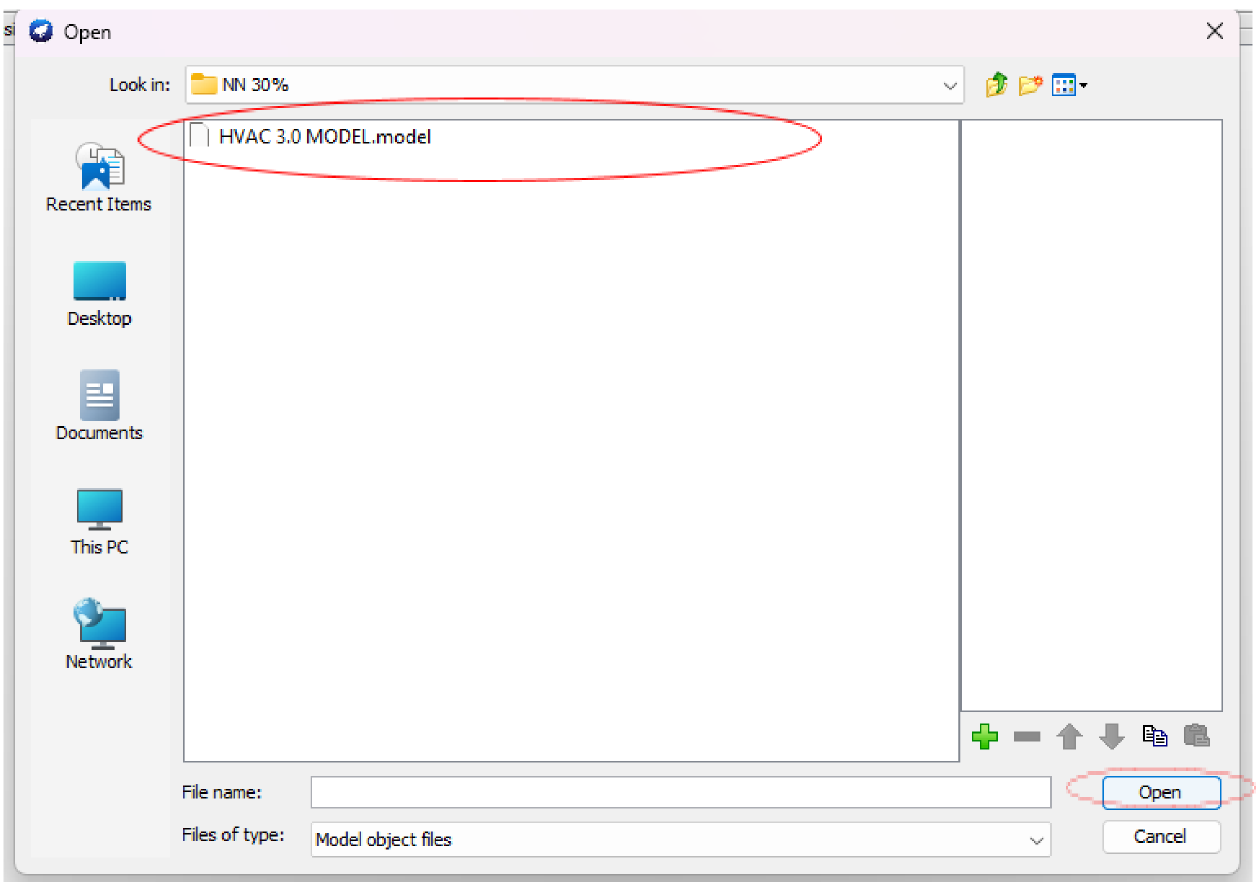


3. This file arff already preprocessed,so at previous does not need to click any button,go to Classify tab (red circle)

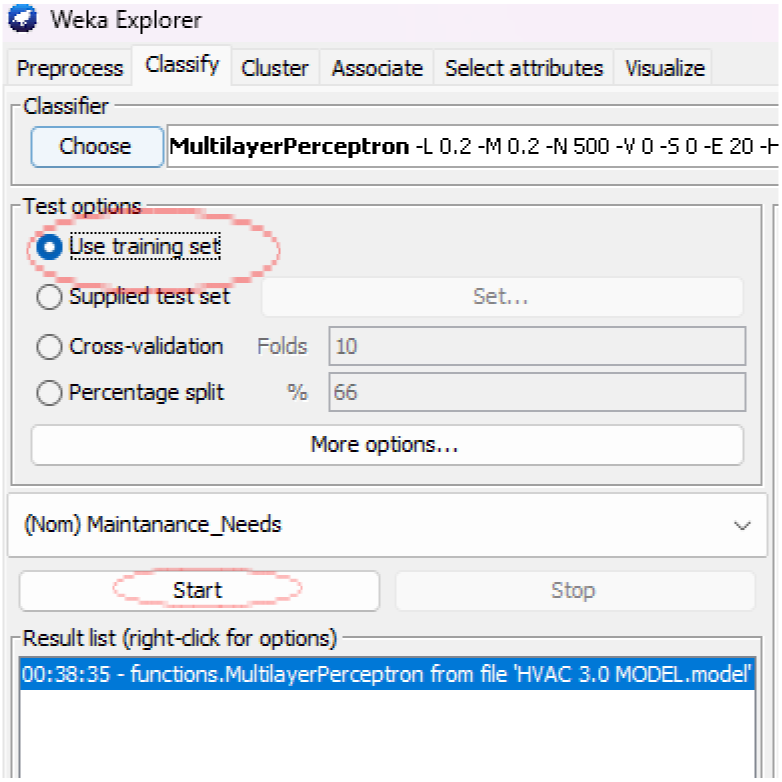


4. Right click on result list then click load model

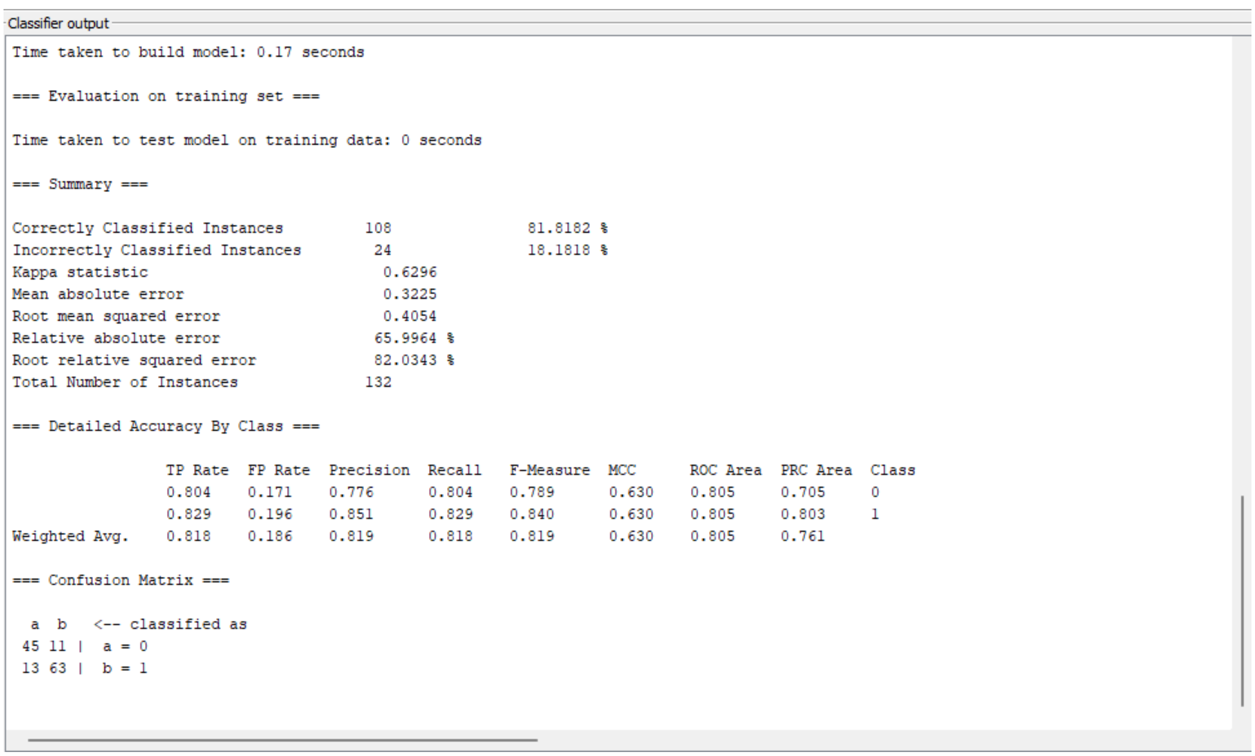




5. Set as below:(red circle)

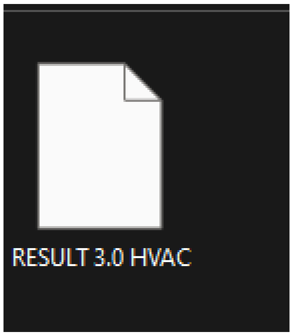


6. After click the red circle;it will output this:



```
Classifier output
Time taken to build model: 0.17 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         108               81.8182 %
Incorrectly Classified Instances        24               18.1818 %
Kappa statistic                          0.6296
Mean absolute error                      0.3225
Root mean squared error                  0.4054
Relative absolute error                 65.9964 %
Root relative squared error             82.0343 %
Total Number of Instances              132

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.804    0.171    0.776      0.804    0.789      0.630    0.805     0.705     0
                 0.829    0.196    0.851      0.829    0.840      0.630    0.805     0.803     1
Weighted Avg.    0.818    0.186    0.819      0.818    0.819      0.630    0.805     0.761

=== Confusion Matrix ===

  a  b   <-- classified as
 45 11 |  a = 0
 13 63 |  b = 1
```

7. If this output cannot be run there is file that save together inside this zip file name as



This file also store the result that after run and it show in txt file.