Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Engineering Cybernetics

Lama Mourad

Predictive Maintenance in Building Automation & HVAC Systems

Master Thesis in Industrial Cybernetics

Master's thesis in Industrial Cybernetics (MIIK)
Supervisor: professor Damiano Varagnolo
July 2023

NTNU
Norwegian University of
Science and Technology

Lama Mourad

Predictive Maintenance in Building Automation & HVAC Systems

Master Thesis in Industrial Cybernetics

Master's thesis in Industrial Cybernetics (MIIK)
Supervisor: professor Damiano Varagnolo
July 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

The master thesis focuses on the implementation of predictive maintenance techniques in building automation and HVAC (Heating, Ventilation, and Air Conditioning) systems. The research aims to develop a model that can predict potential failures and malfunctions in the systems by analysing sensor data and detecting anomalies. The study also examines different approaches to data analysis, such as statistical methods and machine learning algorithms (such as PCA, DBSCAN, K-means, Logistic Regression and the DecisionTreeRegressor). The results demonstrate the effectiveness of predictive maintenance in reducing maintenance costs and improving system reliability and offer insights into the practical implementation of such techniques in real-world scenarios.

Acknowledgements

I would like to express my sincere gratitude to the Norwegian University of Science and Technology – NTNU, and the Department of Industrial Cybernetics for granting me the invaluable opportunity to pursue advanced education, opening doors to a future illuminated with possibilities. I am indebted to the multitude of individuals who played pivotal roles in shaping my journey throughout the course of my master's degree.

First and foremost, I extend my heartfelt appreciation to the professors and instructors who have nurtured my intellectual growth and fuelled my research endeavours.

A special and boundless debt of gratitude is owed to my exceptional supervisor, Professor Damiano Varagnolo, with whom I crossed paths since August 2020 during the enrolment during the course of TTK4225 Systems Theory, Introduction. Professor Damiano's steadfast guidance and unwavering inspiration have been instrumental in shaping the trajectory of my master's thesis. His incisive feedback and constructive criticism have consistently propelled me towards greater heights, fostering significant improvements in the quality and depth of my work. Furthermore, his invaluable arsenal of tools, methodologies, and fresh perspectives have infused my research with novel insights and innovative approaches.

To my cherished family members, my sons Haidara Ali and Mohammed Jawad, and my loving husband Karim, I extend profound gratitude for their unwavering support and Constant motivation throughout my academic pursuit. Their patience during my absence and their understanding of the demanding exam periods and the arduous preparations for my master's thesis have been unwavering. I am forever grateful for their constant encouragement and belief in my abilities.

Finally, my heartfelt appreciation goes out to my parents and friends who have consistently stood by my side, Resolute in their belief in my potential and tirelessly cheering me on throughout my academic journey. Their unwavering faith in me has been a source of profound inspiration, instilling within me the determination to overcome challenges and reach for excellence.

Table of Contents

List of Figures

List of Tables

The key words

HVAC system: Heating, Ventilation, and Air Conditioning system.

BAS: Building automation systems.

PCA: Predictive Maintenance Analysis.

DBSCAN: Density-Based Spatial Clustering of Applications with Noise.

K-Means: K-Means Clustering.

BACnet: Building Automation and Control network.

IoT: Internet of Things.

AMI: Adjusted Mutual Information.

1. Introduction

1.1. Research Background:

1.1.1. Building Automation and HVAC Systems

Building automation and HVAC (Heating, Ventilation, and Air Conditioning) systems play a critical role in ensuring the comfort, health, and safety of building occupants, as well as in maximizing energy efficiency and reducing greenhouse gas emissions (American Society of Heating, 2017).

Building automation refers to the use of technology to control and manage a building's mechanical and electrical systems, including lighting, temperature, ventilation, and security. This can be achieved through the integration of various systems and devices, such as sensors, actuators, and controllers, into a single, unified network (American Society of Heating, 2017).

HVAC systems, on the other hand, are essential components of building automation that regulate temperature and indoor air quality. HVAC systems can consist of various components, including air handling units, boilers, chillers, fans, and ductwork. They are responsible for heating, cooling, and ventilating indoor spaces, as well as for removing excess moisture and controlling air distribution (American Society of Heating, 2017).

The integration of building automation and HVAC systems can lead to significant energy savings, improved indoor air quality, and enhanced comfort for building occupants (American Society of Heating, 2017). For example, building automation systems can optimize HVAC performance by adjusting temperature and ventilation settings based on occupancy patterns and real-time weather conditions. This can reduce energy consumption and minimize the release of greenhouse gases into the atmosphere (American Society of Heating, 2017).

There are several standards and protocols for building automation and HVAC systems, including BACnet (Building Automation and Control Network) (International B. , 2023), KNX (Konnex) (International K. A., 2017), and LonWorks (S.Raji, 1998). These protocols define the communication and interoperability between different devices and systems, ensuring seamless integration and consistent performance (International B. , 2023) (International K. A., 2017) (S.Raji, 1998).

In conclusion, building automation and HVAC systems are essential components of modern buildings, responsible for ensuring the comfort, health, and safety of occupants, as well as for reducing energy consumption and environmental impact.

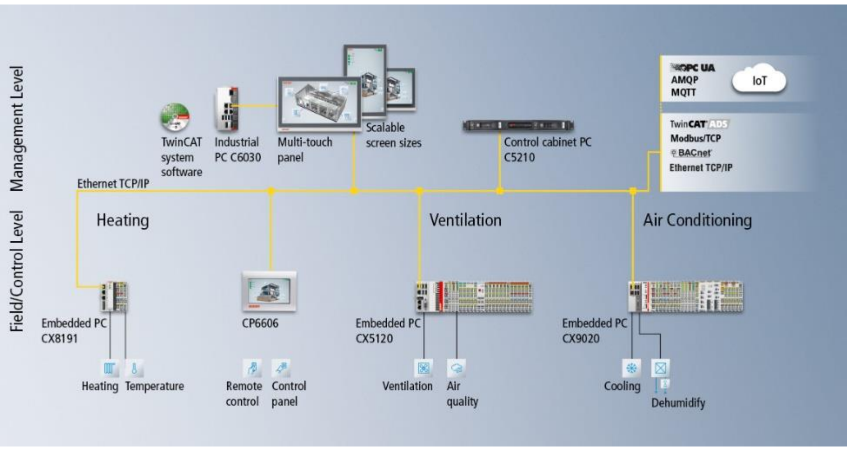


Figure 1: Building Automation/Heating ventilation and Air Conditioning (https://www.beckhoff.com/en-en/industries/building-automation/heating-ventilation-and-air-conditioning/)

1.2. The Motivation

Predictive maintenance is a crucial aspect of building automation and HVAC systems as it can significantly improve their efficiency, reliability, and cost-effectiveness. The following are some of the key benefits of predictive maintenance in building automation and HVAC systems:

1. Reduced Downtime: By predicting equipment failures before they occur, predictive maintenance enables building managers and technicians to perform maintenance proactively, reducing the risk of unexpected failures and minimizing downtime (LLumin, 2023).
2. Improved Equipment Performance: Predictive maintenance can help improve the overall performance of equipment by identifying and addressing potential issues before they cause significant damage or impact equipment efficiency (Ján Drgoňa, 2020).

3. Cost Savings: By reducing downtime, improving equipment performance, and optimizing maintenance schedules, predictive maintenance can help building managers and HVAC technicians save on maintenance costs and improve the cost-effectiveness of building automation and HVAC systems (Butler, 2020).
4. Increased Energy Efficiency: Predictive maintenance can also help increase the energy efficiency of building automation and HVAC systems by identifying and addressing inefficiencies, such as air leaks or poorly functioning equipment (Ján Drgoňa, 2020).

Therefore, predictive maintenance is a vital component of building automation and HVAC systems as it can enhance their efficiency, reliability, and cost-effectiveness, while also reducing downtime and improving energy efficiency.

1.3. The Research Topic of Predictive Maintenance

Predictive maintenance is a maintenance strategy that uses data and analytical models to predict when equipment is likely to fail, allowing maintenance to be performed proactively before failure occurs. The goal of predictive maintenance is to minimize equipment downtime, reduce maintenance costs, and improve overall equipment reliability.

this study is focusing on predictive maintenance in building automation and HVAC systems, specifically examining the use of data to predict problems and failures in filters. By analysing a set of data, it aims to determine if it is possible to discover potential problems before the filters stop working. This type of research can have significant practical applications, as it can help to optimize maintenance schedules and reduce the overall cost of maintenance while improving system performance and reliability.

This is a timely and relevant research topic, as the use of predictive maintenance is growing rapidly in the building automation and HVAC industries. By exploring this topic, I have the opportunity to contribute to the advancement of knowledge in the field and provide valuable insights that can inform future maintenance practices in these systems.

1.4. The Research Questions

1. Can machine learning algorithms be effectively utilized for predictive maintenance in HVAC systems in building automation?
2. Can predictive maintenance in building automation and HVAC systems improve energy efficiency and reduce the frequency of failures?
3. What is the impact of predictive maintenance on the longevity and performance of filters in building automation HVAC systems?
4. How can the implementation of predictive maintenance improve the overall maintenance strategy in building automation and HVAC systems?

1.5. Overview of The Thesis Structure

The structure of the thesis consists of the following chapters:

1. The Introduction.
2. The Literature Reviews.
3. The Methodology.
4. The Results.
5. Discussing the results.
6. The Conclusions.

2. Literature Review

2.1. HVAC System

Heating, ventilation, and air conditioning (HVAC) systems play a crucial role in regulating indoor air quality and maintaining a comfortable temperature in buildings. The following describes the components and processes involved in HVAC systems:

- The HVAC system consists of an exhaust fan and an inlet fan that manage air circulation in the system. Dampers are used to control the airflow by regulating the air velocity and opening and closing the dampers. The system also includes filters that filter the air and cooling coils that modify the air if cooling or heating is required (Ye Yao, 2017).
- The HVAC system typically pumps air in from the outside, which is filtered and mixed with the current air circulation inside the system after passing through the filter detection. When air passes through the coils in the system, it is heated or cooled to the desired temperature for the air channel. The air is then pumped into the air channel by a fan, and then it flows to the zones in the building that require air supply. A reheating coil is installed in each zone, as well as an outlet fan that pumps air out of the space to keep the air moving. The majority of the air drained out of the room is usually mixed with incoming air from the outside, but the same amount of air that entered the air channels is evacuated to the atmosphere at the same time (Selamat, 2020).

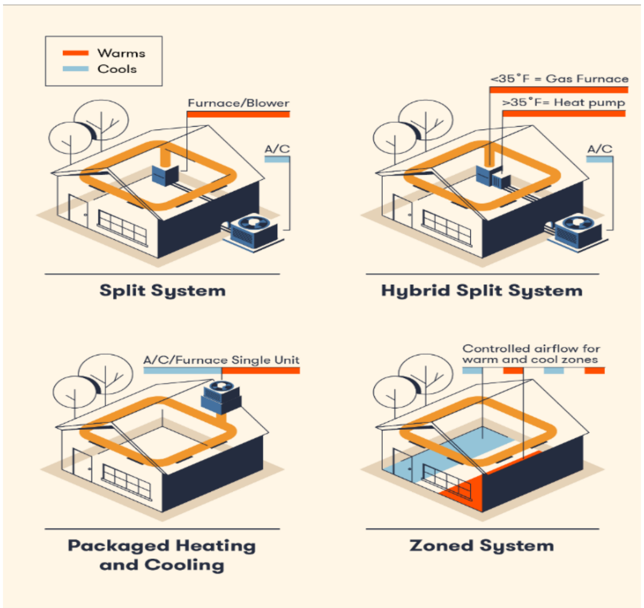


Figure 2: Types of Ducted HVAC Systems (https://www.bigrentz.com/blog/types-of-hvac-systems)

2.1.1. Types of Filters for HVAC Systems

Air filters are essential components of heating, ventilation, and air conditioning (HVAC) systems. They play a crucial role in maintaining indoor air quality by removing harmful particles and pollutants from the air. There are various types of air filters available for HVAC systems, each with unique features and benefits. Here, we will discuss the different types of filters used in HVAC systems and their advantages and disadvantages.

2.1.1.1. Fiberglass Filters:

Fiberglass filters are the most common and affordable type of air filter for HVAC systems. They are made of spun glass fibers and are disposable. They are designed to capture large airborne particles such as dust, dirt, and lint. Fiberglass filters have a low MERV (Minimum Efficiency Reporting Value) rating, which means they are not very effective at capturing smaller particles like pollen and pet dander (Agency, Guide to Air Cleaners in the Home, 2023).

2.1.1.2. Pleated Filters:

Pleated filters are a step up from fiberglass filters in terms of filtration efficiency. They are made of polyester or cotton paper and have a higher MERV rating than fiberglass filters. Pleated filters can capture smaller particles like pollen, mold spores, and pet dander. They have a larger surface area than fiberglass filters, which means they can hold more contaminants and last longer (LLC, 2023).

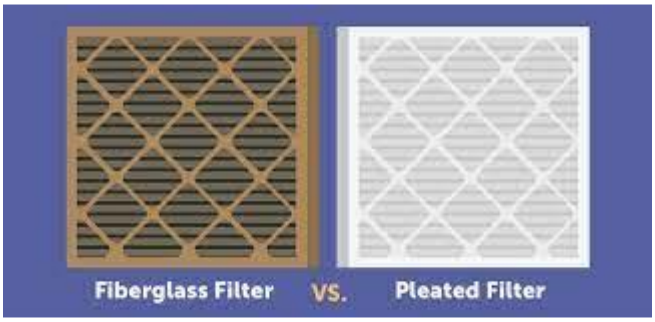


Figure 3:Fiberglass Filters vs Pleated Filters (https://filterking.com/hvac-filters/fiberglass-air-filters-vs-pleated)

2.1.1.3. HEPA Filters:

High-efficiency particulate air (HEPA) filters are the most effective type of air filter for HVAC systems. They are made of densely packed layers of fiberglass and can capture particles as small as 0.3 microns with an efficiency of 99.97%. HEPA filters are commonly used in hospitals, laboratories, and clean rooms to maintain high indoor air quality. However, they require a powerful fan to push air through the dense filter material, which can increase energy costs (Agency, U.S. Environmental Protection Agency, 2023).

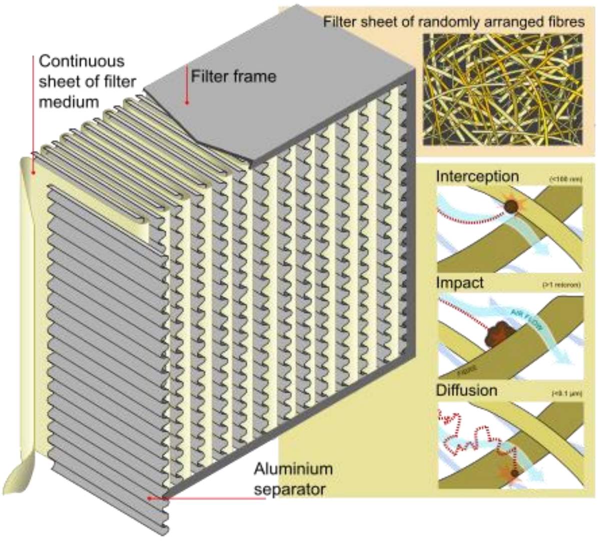


Figure 4:HEPA Filters (https://stellaraircleaning.co.uk/hepa-air-purifier/)

2.1.1.4. Electrostatic Filters:

Electrostatic filters use an electrostatic charge to attract and capture airborne particles. They are made of multiple layers of polypropylene fibers and have a MERV rating between 8 and 12. Electrostatic filters are washable and reusable, which makes them more eco-friendly than disposable filters. However, they can produce ozone and may not be suitable for people with respiratory problems (Lower, 2023).

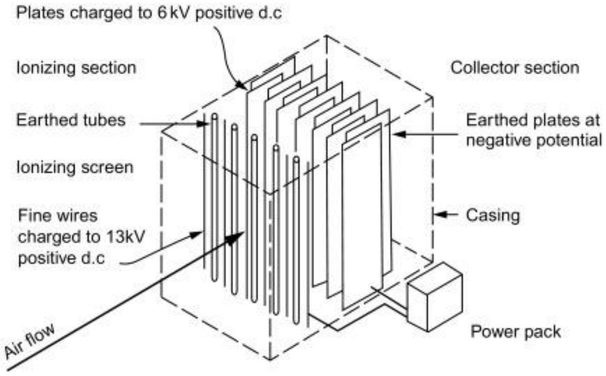

Plates charged to 6 kV positive d.c
Ionizing section
Earthed tubes
Ionizing screen
Fine wires charged to 13kV positive d.c
Air flow
Collector section
Earthed plates at negative potential
Casing
Power pack

Figure 5: Electrostatic Filters (https://www.sciencedirect.com/topics/engineering/electrostatic-filter)

It's important to choose the right filter type for the HVAC system to ensure effective filtration and maintain good indoor air quality.

2.2. Building Automation Systems (BAS)

Building automation systems (BAS) are computer-based control systems that are used to manage and monitor building operations such as heating, ventilation, air conditioning, lighting, and security ((MACC), 2022). These systems are designed to increase energy efficiency, reduce costs, and improve building performance.

There are several components that make up a BAS, including sensors, controllers, and user interfaces (Urvashi, 2018). The sensors are used to measure various building parameters such as temperature, humidity, and occupancy, while the controllers use this information to make decisions about how to adjust the building systems for optimal performance ((MACC), 2022). The user interface is used to allow building managers and occupants to interact with the BAS and make adjustments as needed (Urvashi, 2018).

BAS can be integrated with other building systems such as fire alarms, elevators, and access control systems, creating a comprehensive building management solution ((MACC), 2022).

The use of BAS has been shown to have significant benefits for building owners and operators, including energy savings, reduced maintenance costs, and improved occupant comfort and productivity (American Technical Publishers, 2009).

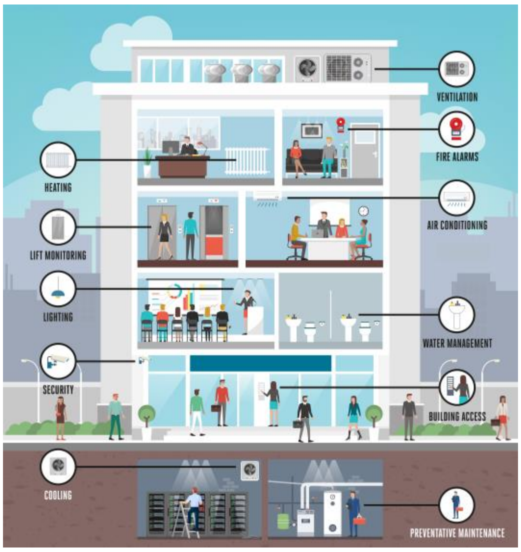


Figure 6: Building Automation System (BAS) (https://www.atalianservest.co.uk/your-guide-to-building-automation-systems/building-automation-systems-bas-infographic-2/)

2.3. Maintenance Strategies and Techniques

Maintenance strategies and techniques refer to the various approaches and methods used to ensure that equipment, machinery, and systems are kept in good condition and working efficiently (Kelly, 2006). The main goal of maintenance is to minimize downtime, prevent breakdowns, and extend the lifespan of the equipment (Altomonte, 2022).

There are several maintenance strategies and techniques, including:

1. Preventive maintenance - involves regularly scheduled maintenance tasks to prevent equipment failure and reduce the likelihood of unplanned downtime (Mobley, 2013).
2. Predictive maintenance - uses data and analytics to identify potential problems before they occur, enabling maintenance teams to take proactive measures to prevent failures (Kelly, 2006).
3. Corrective maintenance - addresses problems that have already occurred, with the goal of restoring the equipment to its normal operating condition (Mobley, 2013).
4. Condition-based maintenance - relies on sensors and other monitoring equipment to track the condition of equipment and identify maintenance needs based on its actual condition (Kelly, 2006).
5. Total productive maintenance - aims to involve all employees in the maintenance process and create a culture of continuous improvement to enhance the overall efficiency of the organization (Altomonte, 2022).

Some common maintenance techniques include lubrication, cleaning, calibration, inspection, and repair (Mobley, 2013).

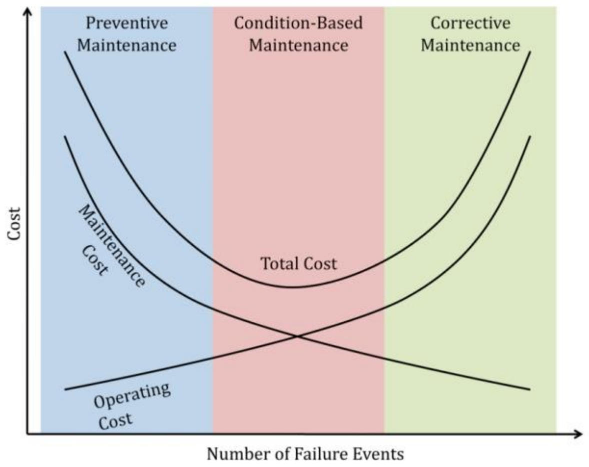


Figure 7:Operating and Maintenance Cost Chart (https://www.researchgate.net/figure/Operating-and-Maintenance-Cost-Chart-3_fig1_228749201)

2.4. Predictive Maintenance Concept and Tools

Predictive maintenance is a maintenance strategy that utilizes data analysis tools and techniques to predict equipment failure and proactively perform maintenance to prevent downtime. The primary objective of predictive maintenance is to decrease maintenance costs, enhance equipment reliability and availability, and improve overall operational efficiency.

There are several key concepts involved in predictive maintenance:

- Machine learning algorithms, such as decision trees, random forests, and gradient-boosted trees, can analyse vast amounts of sensor data to detect patterns and anomalies that may indicate an impending failure (Mahsa Shoaran, 2018). These algorithms can also predict the remaining useful life (RUL) of a machine, which can help schedule maintenance activities.
- Condition-based monitoring is another technique that involves monitoring the condition of a machine in real-time and comparing it against a baseline to identify any deviation that may indicate a potential problem. The data collected can be used to build predictive models that flag anomalies and provide early warnings of possible failures (Nita Yodo, 2022).
- Predictive analytics combines machine learning, data mining, and statistical analysis to forecast future outcomes. Predictive analytics can analyze

operational data, sensor data, and maintenance history to develop predictive models that identify potential failures before they occur (IBM, 2023).

- The Internet of Things (IoT) and wireless sensing technologies provide the means to collect vast amounts of real-time data from sensors, machines, and equipment. This data can be used to develop predictive models that provide early warnings of potential failures (Yongyi Ran, 2019).
- Artificial neural networks are another type of machine learning algorithm that can analyse large amounts of data to detect patterns and anomalies. These algorithms can be used to detect anomalies in sensor data that may indicate an impending failure (Mahsa Shoaran, 2018).

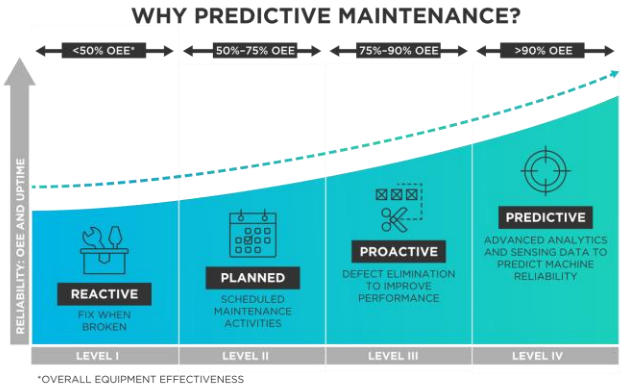

WHY PREDICTIVE MAINTENANCE?
<50% OEE*
50%-75% OEE
75%-90% OEE
>90% OEE
RELIABILITY: OEE AND UPTIME
REACTIVE
FIX WHEN BROKEN
PLANNED
SCHEDULED MAINTENANCE ACTIVITIES
PROACTIVE
DEFECT ELIMINATION TO IMPROVE PERFORMANCE
PREDICTIVE
ADVANCED ANALYTICS AND SENSING DATA TO PREDICT MACHINE RELIABILITY
LEVEL I
LEVEL II
LEVEL III
LEVEL IV
*OVERALL EQUIPMENT EFFECTIVENESS

Figure 8:Predictive Maintenance (https://www.tibco.com/reference-center/what-is-predictive-maintenance)

2.5. Machine Learning

Machine learning refers to a type of artificial intelligence that involves training a computer to learn from data, rather than being explicitly programmed (Ian Goodfellow, 2016). The idea behind machine learning is to improve a computer's performance on a specific task by using examples and experiences to teach it, rather than relying on a fixed set of rules. At the core of machine learning is the concept of a model, which is a mathematical representation of the relationship between input data and output data. The model is trained on a set of labelled data, known as the training set, and learns to make predictions on new, unseen data, known as the test set.

There are several types of machine-learning models, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model learns to map input data to output data by using a set of labelled examples. Popular algorithms used in supervised learning include decision trees, random forests, and neural networks (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009). On the other hand, unsupervised learning involves discovering patterns in data without any labelled examples. Common unsupervised learning algorithms include k-means clustering, principal component analysis, and autoencoders. Finally, reinforcement learning involves an agent learning to take actions in an environment to maximize a reward signal. Reinforcement learning is often used in applications such as robotics and game playing. Popular algorithms used in reinforcement learning include Q-learning, policy gradient methods, and actor-critic methods (Sutton, 2018).

There are several programming languages and libraries that are commonly used in machine learning, including Python, R, TensorFlow, PyTorch, and Scikit-learn. These tools provide a variety of algorithms and techniques for building and training machine learning models.

Machine learning is a powerful tool for building intelligent systems that can learn and improve over time. By understanding the fundamental concepts and techniques of machine learning, developers and data scientists can create robust and efficient models that can be applied to a wide range of applications.

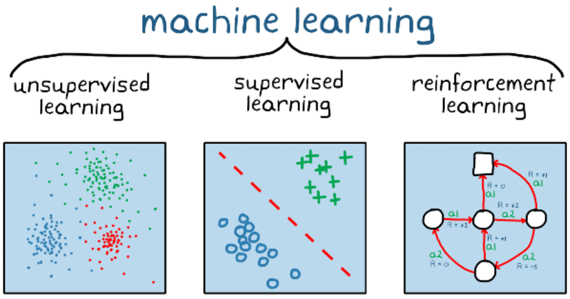


Figure 9: Machine Learning(https://cloud2data.com/types-of-machine-learning/)

2.6. Choosing Machine Learning

2.6.1. Significance of Data

Machine learning refers to a type of artificial intelligence that involves training a computer to learn from data, rather than being explicitly programmed (Goodfellow, Deep feedforward networks, 2016). The idea behind machine learning is to improve a computer's performance on a specific task by using examples and experiences to teach it, rather than relying on a fixed set of rules. At the core of machine learning is the concept of a model, which is a mathematical representation of the relationship between input data and output data. The model is trained on a set of labelled data, known as the training set, and learns to make predictions on new, unseen data, known as the test set.

There are several types of machine-learning models, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model learns to map input data to output data by using a set of labeled examples. Popular algorithms used in supervised learning include decision trees, random forests, and neural networks (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009).

On the other hand, unsupervised learning involves discovering patterns in data without any labeled examples. Common unsupervised learning algorithms include k-means clustering, principal component analysis, and autoencoders. Finally, reinforcement learning involves an agent learning to take actions in an environment to maximize a reward signal. Reinforcement learning is often used in applications such as robotics and game playing. Popular algorithms used in reinforcement learning include Q-learning, policy gradient methods, and actor-critic methods (Sutton, 2018).

2.6.2. Requirements for Implementation of Machine Learning

In order to implement Machine Learning projects, businesses must conduct a business understanding investigation to determine the potential profitability of data mining (Hannun, 2019). Machine Learning business solutions are typically successful in two problem types: demand prediction and sufficient isolation from outside interference. Demand prediction focuses on predicting new data based on previous data, rather than seeking answers in the correlation between the data, making it necessary to have a large sample size of data (Hannun, 2019). The second problem type requires the isolation of data as the learning algorithm relies on the factors included in the problem. If these factors change, the algorithm can no longer match the new information with prior patterns, causing it to be unable to process the new data (Hannun, 2019).

Examples of business applications for Machine Learning include content recommendations on YouTube, autonomous driving, marketing, and machine automation (Kourou, 2015) (Libbrecht, 2015) (Witten, Data mining: practical machine learning tools and techniques with Java implementations, 2002).

2.6.3. Machine Learning Requirements

Machine learning algorithms require high-quality data to perform at a satisfactory level, making insufficient data a significant disadvantage (Witten, Data mining: practical machine learning tools and techniques , 2002). Insufficient data often results from data not being stored primarily, unauthorized data, or data that is too expensive to obtain. A mediocre machine learning algorithm supported by big data usually outperforms a superior algorithm with an inferior dataset. Thus, having a big sample size of data is the most important requirement for a machine learning model (Witten, Data mining: practical machine learning tools and techniques, 2002).

2.7. Outliers

To gain a better understanding of detecting outliers, it's important to define what constitutes an outlier and its characteristics. Outliers are patterns in a dataset that deviate significantly from its expected behaviour. These anomalies can arise due to various factors such as the nature of the data and the operations performed on it (Chandola, 2009). By understanding the underlying causes of outliers, one can employ effective techniques for identifying and managing them in data analysis.

2.7.1. Methods for Detecting Anomalies

Different types of anomaly detection techniques will be covered in this chapter, along with a discussion of their advantages and disadvantages. The choices selected for this thesis will be based on the denouement. The following two sections serve as broad categories for anomaly detection techniques based on various machine learning algorithms.

2.7.1.1. Supervised Machine Learning Methods

Supervised learning is a type of machine learning that involves training a model on labelled data, with the goal of enabling the model to make predictions on new, unlabelled data. This type of learning is widely used in applications such as image recognition, speech recognition, and natural language processing (G{\"o}nen, 2010) (Goodfellow, Deep learning, 2016) (Kelleher, 2018) (Shalev-Shwartz, 2014). There are several types of supervised learning methods, including regression and classification.

Table 1: Advantages of Supervised Learning

Advantage	Explanation
Accuracy	Supervised learning methods tend to produce accurate predictions and classifications, especially when the training data is of high quality (G{\"o}nen, 2010).
Versatility	These methods can be applied to a wide range of problems, including image classification, speech recognition, and natural language processing (Goodfellow, Deep learning, 2016).
Ease of Interpretation	The results of supervised learning methods are often easy to interpret, which can be useful for making informed decisions (Kelleher, 2018).
Reduced Data Requirements	With supervised learning, it is often possible to achieve high accuracy with a smaller amount of data compared to unsupervised learning methods (Shalev-Shwartz, 2014).

Table 2: Disadvantages of Supervised Learning

Disadvantage	Explanation
Need for Labeled Data	Supervised learning methods require labeled data for training, which can be time-consuming and costly to obtain (G{\"o}nen, 2010).
Bias and Overfitting	Supervised learning models can become overly specialized to the training data, resulting in overfitting and reduced generalization capabilities (Goodfellow, Deep learning, 2016).
Limited Scope	Supervised learning methods are limited to the specific classes or outputs that are included in the training data (Kelleher, 2018).
Difficulty with Outliers	Supervised learning models may struggle to accurately classify outliers or unusual data points (Shalev-Shwartz, 2014).

2.7.1.2. Unsupervised Machine Learning Methods

Unsupervised learning is a type of machine learning where the algorithm is not given any labelled data to train on, but instead must find patterns and structures within the data on its own. The goal of unsupervised learning is to identify relationships, groupings, and similarities in data without being given any pre-defined categories or classes.

There are several different methods of unsupervised learning, including:

1. Clustering: Clustering involves grouping data points together based on their similarity to one another. This can be useful for identifying patterns or groups within a dataset (Devanathan, 2021).
2. Dimensionality Reduction: This technique involves reducing the number of features or variables in a dataset while still retaining as much information as possible. This can be useful for visualizing high-dimensional data or reducing the computational complexity of a model (Nielsen, 2019) (VinayakGoyal, 2021).
3. Anomaly Detection: Anomaly detection involves identifying outliers or anomalies in a dataset, which may represent errors or unusual events (Laskin, 2021).

Here is a table (3) outlining the advantages and disadvantages of unsupervised learning (Devanathan, 2021) (Laskin, 2021):

Table 3: Advantages and Disadvantages of Unsupervised Learning

Advantages	Disadvantages
Can identify patterns and structures in data that may not be apparent through manual inspection.	Results can be difficult to interpret, as the algorithm may identify patterns that are not meaningful or relevant.
Can be useful for discovering new categories or groupings within a dataset.	Results can be sensitive to the choice of algorithm and parameters.
Does not require labeled data, which can be expensive or time-consuming to obtain.	Can be computationally intensive and may require significant processing power.
Can be used for a wide range of applications, from data mining to image and speech recognition.	May not perform as well as supervised learning methods when labeled data is available.

One popular unsupervised learning technique is dimensionality reduction, such as Principal Component Analysis (PCA), and Some of the most popular unsupervised learning methods include clustering and nearest neighbour-based methods (Devanathan, 2021) (Nielsen, 2019) (Laskin, 2021).

2.7.2. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used unsupervised learning technique for reducing the dimensionality of a dataset while retaining as much of the variation as possible. It involves transforming the original variables into a new set of uncorrelated variables called principal components, which are ordered by the amount of variance they explain in the data. By reducing the number of variables, PCA can simplify the data and make it easier to analyse, visualize, and model (Devanathan, 2021) (Nielsen, 2019) (Laskin, 2021).

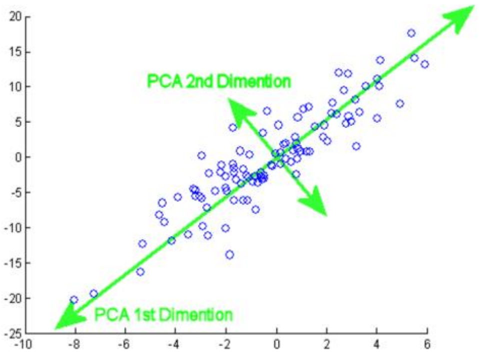


Figure 10: Principal Components Analysis (https://programmathically.com/principal-components-analysis-explained-for-dummies/)

The table (4), describes the advantages and disadvantages of Principal Component Analysis (Mangla, 2022) (Karunakaran, 2022) (Zhang Y.-P. a.-H.-T.-H., 2019).

Table 4: Advantages and Disadvantages of PCA

Advantages of PCA	Disadvantages of PCA
Simplifies data and makes it easier to analyze and visualize.	Can lead to loss of information if too many principal components are discarded.
Can reduce the risk of overfitting in machine learning models by reducing the number of variables.	Interpretation of principal components can be difficult, especially when there are many variables involved.
Can reveal hidden relationships and correlations between variables in a dataset.	Results can be sensitive to the choice of parameters, such as the number of principal components to retain.
Can speed up the computation time for algorithms that use the transformed data.	

2.7.3. Clustering Methods

Clustering is a type of unsupervised learning method that can be used for anomaly detection. This approach involves comparing new logs with defined normal data instances labelled as clusters (Mishra, 2017).

There are two ways to approach anomalies with cluster analysis. The first way is to cluster the data set and analyse the frequency of each cluster. Normal logs are considered to be part of the frequent large data set, while abnormal logs are located in inadequate clusters. The second approach is to assume that normal logs are located closer to the center than anomalous logs, which are assumed to be located further from the center (Mishra, 2017).

In terms of testing, clustering analysis is considered the fastest unsupervised learning method in the testing phase compared to other methods (MengYang Liu, 2022). However, the interpretation of the results can be difficult, especially when the number of clusters is not predetermined. Also, the results can be sensitive to the choice of algorithm and its parameters (Beeck, 2021).

machine-learning/Clustering methods have several advantages and disadvantages, as shown in table (5), including (Jain, 1999) (Shukor, 2015) (Tan, Data, 2006):

Table 5: Advantages and Disadvantages of Clustering Methods

Advantages	Disadvantages
Does not require labeled data, which can be expensive and time-consuming to obtain.	Results can be sensitive to the choice of algorithm and its parameters.
Can be used for exploratory data analysis to identify hidden patterns and structures.	Interpretation of results can be difficult, especially when the number of clusters is not predetermined.
Can be useful for segmentation and targeting in marketing and customer relationship management.	Scalability can be an issue when dealing with large datasets.
Can detect anomalous data points that do not fit into any cluster, which can be useful for outlier detection and fraud detection.	Can be influenced by outliers and noise in the data.

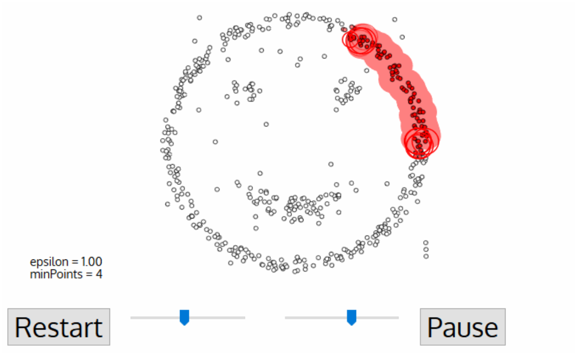
2.7.3.1. DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm used in unsupervised machine learning. Unlike other clustering algorithms, it does not require specifying the number of clusters in advance. Instead, it identifies clusters based on the density of points in the data space (Sander, 1998).

The algorithm starts by randomly selecting a point in the data set and determining the density of the points around it within a specified radius. If the density of points is above a certain threshold, the point is considered a core point and is added to a cluster. The algorithm then finds all the neighbouring points of the core point and adds them to the same cluster if their density is also above the threshold. This process is repeated until all points in the cluster are identified (Sander, 1998).

DBSCAN can also identify noise points, which are points that do not belong to any cluster. These points are either too far from any core point or have a density below the threshold (Sander, 1998).

One of the advantages of DBSCAN is its ability to handle arbitrary-shaped clusters and noisy data. It is also computationally efficient and does not require the user to specify the number of clusters in advance. However, it can be sensitive to the choice of the distance metric and parameters used to define density, which can affect the results (Han, 2022) (Tan, Data, 2006).



epsilon = 1.00
minPoints = 4

Figure 11:DBSCAN Clustering Algorithm (https://www.datacamp.com/blog/clustering-in-machine-learning-5-essential-clustering-algorithms)

Figures (11 and 12) describe how the DBSCAN algorithm works.

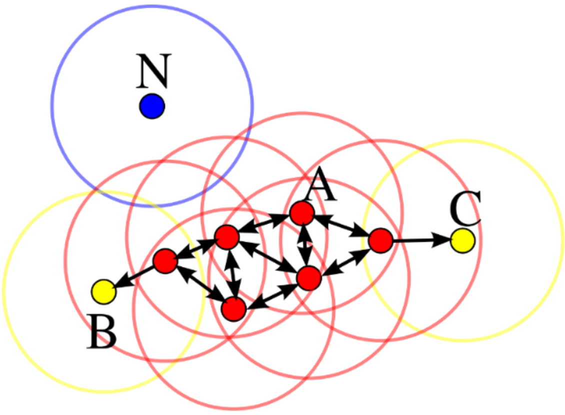


Figure 12: DBSCAN-Illustration (https://jcsites.juniata.edu/faculty/rhodes/ml/dbscan.htm)

2.8. The Existing Knowledge

Predictive maintenance (PdM) is a maintenance strategy that aims to predict and prevent equipment failures before they occur. In building automation and HVAC systems, PdM can help reduce maintenance costs, improve system reliability, and minimize downtime.

Recent studies have demonstrated the feasibility of using predictive maintenance in building automation and HVAC systems. For example, in a study by (Celina Gómez, 2019), a predictive maintenance model was developed using IoT sensors and machine learning techniques, and applied to a heating, ventilation, and air conditioning (HVAC) system in a building. The results showed that the model could accurately predict equipment failures, reducing the number of unplanned breakdowns and maintenance costs.

Another study by (Klein, 2019) proposed a predictive maintenance framework for building automation systems, which uses data from IoT sensors and applies machine learning techniques to predict equipment failures. The study showed that the framework could significantly improve the reliability of building automation systems, reducing downtime and maintenance costs.

These studies demonstrate the potential of predictive maintenance for building automation and HVAC systems. However, further research is needed to develop more sophisticated predictive models and to test their performance in different types of buildings and HVAC systems.

2.9. Results From Previous Studies:

Predictive maintenance has been widely studied in the context of building automation and HVAC systems in the past few decades. Here are some of the key findings and references to some of the relevant studies:

Machine learning algorithms have been shown to be effective in predicting equipment failures in building automation and HVAC systems. For example, a study by (Wenlong Fu, 2018) found that Random Forest and Support Vector Machines (SVM) algorithms were able to accurately predict equipment failures with a high degree of accuracy.

The use of sensors and data analytics can significantly improve the accuracy of predictive maintenance in building automation and HVAC systems. A study by (Xianfei Yin, Building information modelling for off-site construction: Review and future directions, 2019) found that the use of sensors and data analytics was able to improve the accuracy of predictive maintenance by up to 40%.

Predictive maintenance can lead to significant cost savings in building automation and HVAC systems. A study by (Dzhambazov, 2017) found that the use of predictive maintenance in building automation and HVAC systems can lead to cost savings of up to 30% compared to traditional maintenance methods.

Predictive maintenance can also improve the overall efficiency of building automation and HVAC systems. A study by (Li, 2021) found that predictive maintenance can lead to an increase in energy efficiency by up to 20%.

These studies demonstrate the potential benefits of using predictive maintenance in building automation and HVAC systems, including improved accuracy, cost savings, and increased efficiency.

2.10. Conclusion

Predictive maintenance has been shown to be an effective strategy for reducing maintenance costs, improving system reliability, and minimizing downtime in building automation and HVAC systems. Recent studies have demonstrated the feasibility of using predictive maintenance in these systems and have shown that machine learning algorithms, sensors, and data analytics can significantly improve its accuracy (Jun

Yuan, 2020) (Xianfei Yin, Building information modelling for off-site construction: Review and future directions, 2019) (Dzhambazov, 2017).

Moreover, the use of predictive maintenance can lead to significant cost savings and increase the overall efficiency of building automation and HVAC systems. However, further research is needed to develop more sophisticated predictive models and to test their performance in different types of buildings and HVAC systems (Chen, 2021).

Additionally, challenges related to data collection, data quality, and data integration must be addressed to fully realize the potential of predictive maintenance in building automation and HVAC systems (Ashtari Talkhestani, 2019).

In summary, predictive maintenance is a promising approach to improving the performance of building automation and HVAC systems, and continued research in this area can lead to even more significant benefits in the future.

3. Methodology

3.1. Collection of Initial Data

3.1.1. Importance of Data Collection

The purpose of the data collection process in this research on predictive maintenance in building automation and HVAC systems is to gather relevant and accurate information that can be used to develop and validate predictive models. The data collected will be used to analyse patterns and relationships between various factors, such as pressure, airflow, and equipment performance, in order to identify the most critical indicators of potential equipment failure.

The significance of the data collection process lies in its ability to provide a foundation for the predictive maintenance approach, which is crucial for optimizing the maintenance of HVAC systems in building automation. By collecting data on equipment performance and other relevant variables, the predictive maintenance approach can help minimize downtime, reduce maintenance costs, and improve the overall efficiency and reliability of building automation systems (Gbadamosi, 2019).

Moreover, the data collected will also help to identify any potential issues and hazards associated with HVAC systems, allowing for proactive maintenance and repair before they become major problems (Zhang W. a., 2020).

Therefore, the data collection process plays a crucial role in the success of the predictive maintenance approach and is a crucial step in ensuring the optimal performance of building automation and HVAC systems (Kuo, 2019) (Ke, 2020).

3.1.2. Data Sources

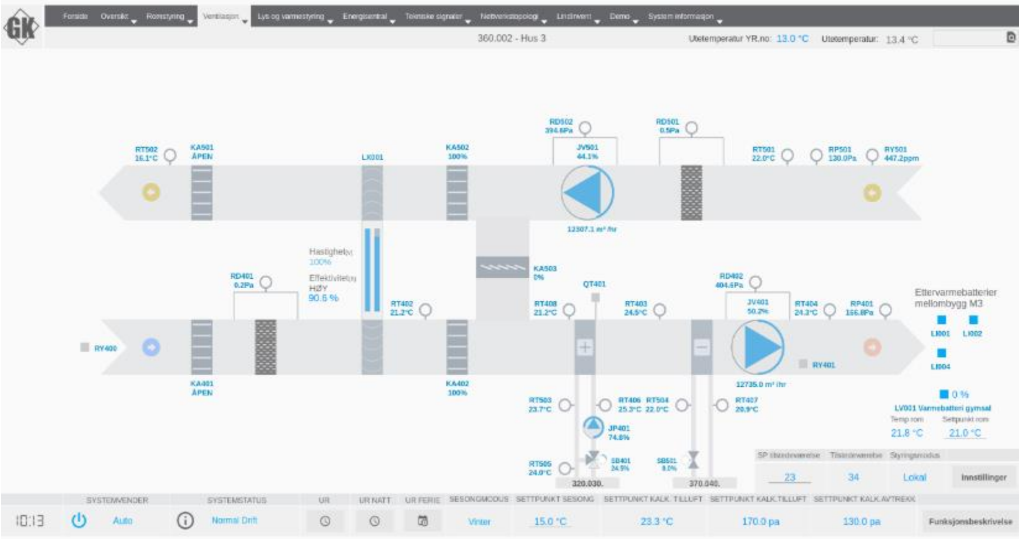
The primary data for this study was obtained from Piscada company, an industry leader. The dataset provided by Piscada was the foundation of my research, and it was analysed and interpreted to draw meaningful insights and conclusions.

GK Cloud is specifically designed for the operation, monitoring, and analysis of building-based technical installations. Operating data can be obtained from virtually any type of SD facility or monitoring system and collected in a single interface accessible from a PC, tablet, or mobile device using Piscada's technology platform.

IoT and sensors have aided in the monitoring of everything from pumps, ventilation systems, lighting control, and access control systems to temperatures, humidity, and sudden voltage drops via apps. However, real gains can be realized only when all of this is brought together in a clear overall picture. According to Per Arve Ekle, divisional director, and head of GK Byggautomasjon, "GK Cloud makes it easy to keep track of all technology in construction and contributes to predictable, stable, and energy-efficient operation of our customers' technical facilities." (GK, 2020)

3.2. Data Description

The information gathered came from seven ventilation systems, all of which were built up similarly to what is shown in the diagram.



Temp rom 21.8 °C
Settpunkt rom 21.0 °C

SP tilandsvarme 23
Tilandsvarme 34
Styringsmodus Lokal
Innstillinger

10:13
SYSTEMVENDER Auto
SYSTEMSTATUS Normal Drift
UR
UR NATT
UR FERIE
SESONGMODUS Vinter
SETTPUNKT SESONG 15.0 °C
SETTPUNKT KALK. TILLUFT 23.3 °C
SETTPUNKT KALK.TILLUFT 170.0 pa
SETTPUNKT KALK.AVTREKK 130.0 pa
Funksjonsbeskrivelse

Figure 13: The HVAC System of my Study

We have received a large unsupervised dataset from the Piscada company, containing seven pickle files that cover the period from June 2019 to October 2021. The dataset includes data from seven HVAC systems, each identified by a unique symbol (001, 002, 006, 007, 008, 009, 011). Each file contains data from three years (2019, 2020, and 2021) and each year consists of (144624) rows of observations from ten sensors.

The sensors in each file include the pressure of the filter in both supply and return ducts, the pressure of the fan in both supply and return ducts, the percentage of the damper or valve in the fan in both supply and return ducts, and the flow of air in both supply and return ducts, the airflow is calculated by multiplying the pressure of the fan by the k-factor.

Table 6: Description of Data Set in Each File

(The system, PKL File)	Row Counts	The size
001	164.273.662	289.282.948 KB
002	134.976.733	243.739.95 KB
006	259.118.856	516.626.190 KB
007	338.333.619	597.239.070 KB
008	71.295.623	123.699.983 KB
009	328.761.551	572.925.120 KB
011	95.031.660	169.907.403 KB

The sensors in each file are explained in the table (7).

Table 7: The Variables in Each File of Dataset

Sensor symbol	The description	Suffix indication	Measuring unit
JV401_C JV501_C	a vane sensor used to measure the opening angle of a damper or a valve in both supply and return ducts in an HVAC system	C: control D: Drift	Expressed as a percentage
JV401_D JV501_D	On/off status for fan in both supply and return ducts in an HVAC system		
RD401_MV RD501_MV	a pressure sensor used to monitor filter sensor pressure in both supply and return ducts in an HVAC system	MV: measurement value	(Pa)
RD402_MV RD502_MV	a pressure sensor used to monitor fan sensor pressure in both supply and return ducts in an HVAC system		

RF401_KV RF501_KV	a sensor that uses the pressure value in the fan multiplied by the k-factor, it can be used to monitor fan performance and measure the pressure drop across the fan.	KV: calculation value	(m³/s)

Where the sensors which have a number (4) indicates to the supply duct and the sensors which have a number (5) indicates to the return duct.

The following plots are representing each sensor original observation values over the time.

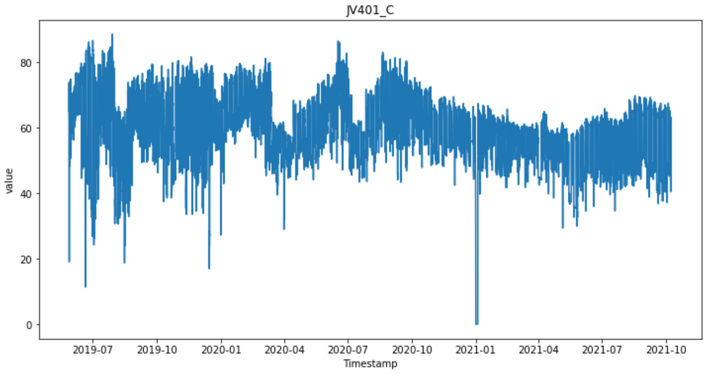


Figure 14:The Percentage Value of the Fan Valve Over Time in The Supply Duct

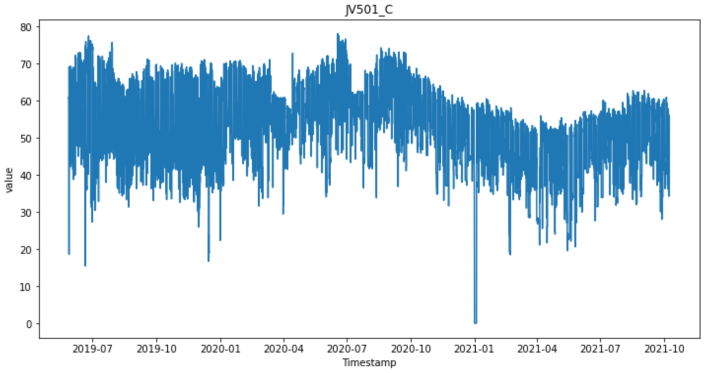


Figure 15::The Percentage Value of the Fan Valve Over Time in The Return Duct

In the analysis of the HVAC system, one of the observed sensors pertains to the percentage value of the fan valve. The recorded observations indicate that the sensor readings exhibit oscillatory behaviour, centring around a value of approximately 60 from July 2019 to November 2020. However, a distinct and significant decrease in the sensor values is observed from November 2020 until the end of October 2021.

This observation highlights a notable temporal pattern in the behaviour of the fan valve, suggesting a potential change or adjustment in the HVAC system during this period. The observed decrease in the sensor values indicates a deviation from the previous oscillatory pattern, potentially indicating a shift in the operational dynamics of the fan valve.

Also, it was observed that certain sensor readings exhibited outlier values. These outlier values deviated significantly from the expected range of observations and could potentially impact the overall analysis and interpretation of the data.

The presence of outliers in the sensor readings introduces a challenge in accurately capturing the underlying patterns and trends within the HVAC system. We will see later in this paper how these anomalies will be explained.

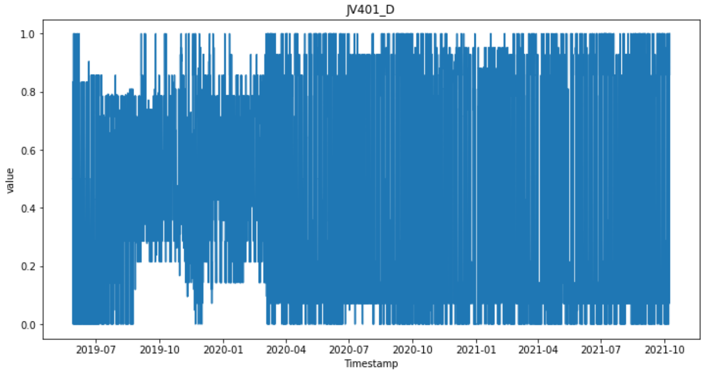


Figure 16: On/Off Status of The Fan Over Time in The Supply Duct

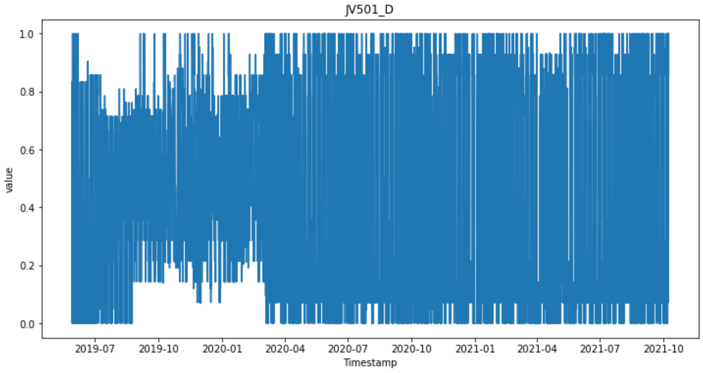


Figure 17: On/Off Status of The Fan Over Time in The Return Duct

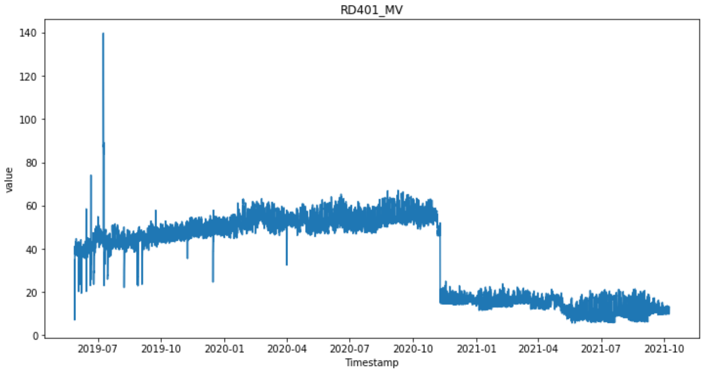


Figure 18: The Filter Pressure Over Time in The Supply Duct

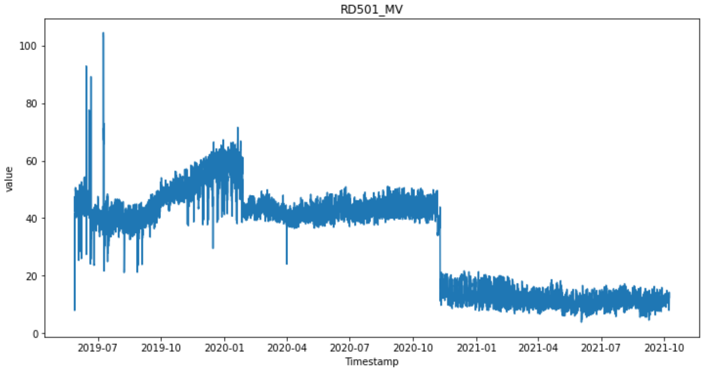


Figure 19: The Filter Pressure Over Time in The Return Duct

In the analysis of the HVAC system, the second sensor under consideration corresponds to the filter pressure. The sensor readings exhibit distinct temporal patterns over the observed time period. Specifically, two clear patterns are observed: the first pattern spans from July 2019 to November 2020, while the second pattern extends from November 2020 until the end of October 2021.

Analysis of the first pattern reveals relatively higher values of filter pressure, indicating a certain level of pressure build up within the filter during this period. This pattern suggests a consistent operational behaviour of the HVAC system, with the filter experiencing relatively higher-pressure levels over time.

In contrast, the second pattern demonstrates lower observations of filter pressure compared to the first pattern. This shift in the observed values indicates a departure from the previous operational behaviour. The lower filter pressure during this period may be indicative of either a change in system conditions or an adjustment in the HVAC system's operational parameters, resulting in reduced pressure build up within the filter.

The identification of these distinct patterns in the filter pressure readings provides valuable insights into the performance and dynamics of the HVAC system. Further analysis and investigation are warranted to understand the factors contributing to the observed changes, including system modifications, maintenance activities, or variations in environmental conditions.

By comprehensively studying and interpreting these patterns, it is possible to gain a deeper understanding of the HVAC system's behaviour and identify potential areas for optimization or improvement.

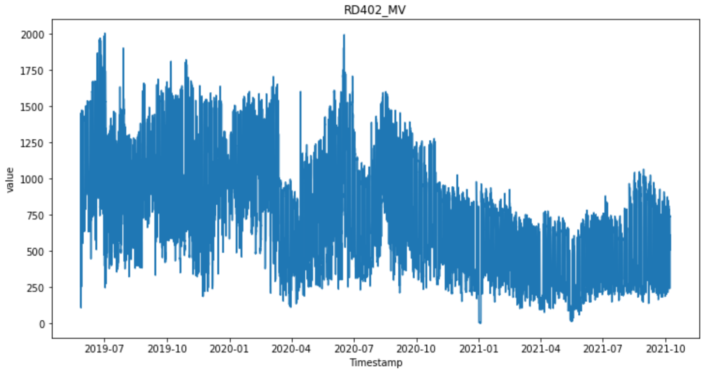


Figure 20: The Fan Pressure Over Time in The Supply Duct

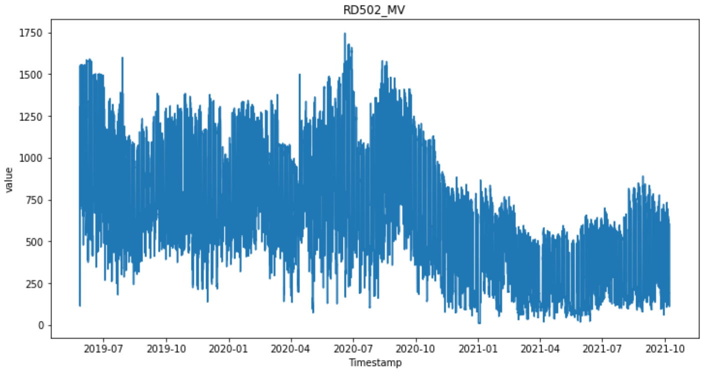


Figure 21: The Fan Pressure Over Time in The Return Duct

We can see here also that the recorded sensor readings exhibit distinct temporal patterns over the observed time period. Two clear patterns are observed: the first pattern spans from July 2019 to November 2020, while the second pattern extends from November 2020 until the end of October 2021.

During the first pattern, the sensor readings demonstrate oscillatory behaviour, with values oscillating around an average of approximately 1000. This consistent oscillation indicates a stable operational state of the fan pressure during this period. In the second pattern, a noticeable shift in the observed values of fan pressure is observed. The values during this period are consistently lower compared to the first pattern. This suggests a significant deviation from the previous operational behaviour of the fan pressure, potentially indicating a change in system dynamics or adjustments made to the HVAC system.

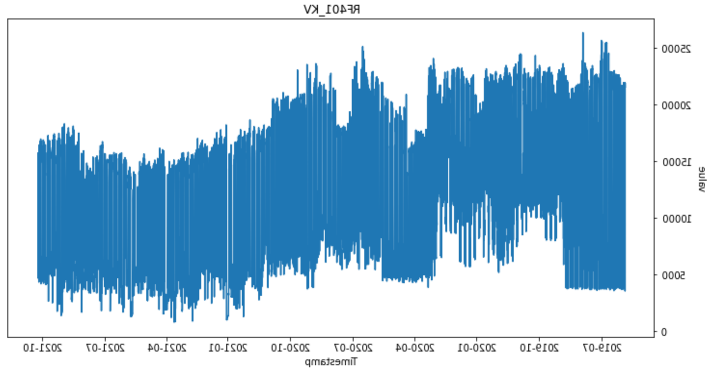


Figure 22: The Airflow Over Time in The Supply Duct

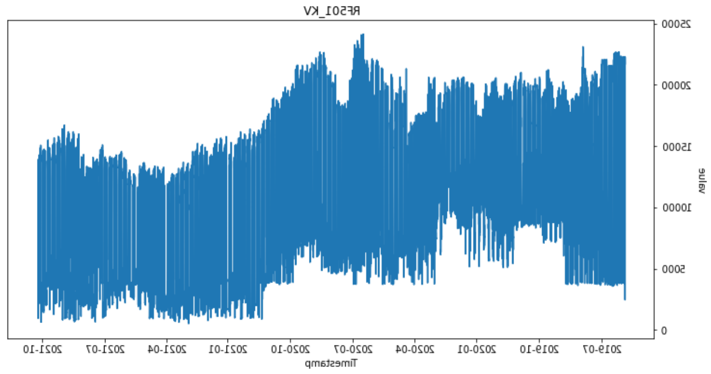


Figure 23: The Airflow Over Time in The Return Duct

Contrary to the previous sensors, the air flow sensor exhibits two distinct patterns over the same time period. The first pattern, spanning from July 2019 to November 2020, is characterized by relatively lower observation values of air flow, whereas the second pattern, from November 2020 until the end of October 2021, shows significantly higher values.

During the first pattern, the air flow sensor readings indicate a relatively lower level of air flow within the HVAC system. This suggests a certain operational state where the air flow was consistently lower during this time frame. In contrast, the second pattern reveals a clear shift in the observed values, with air flow measurements indicating significantly higher levels compared to the first pattern. This change suggests a departure from the previous operational behaviour, with the HVAC system exhibiting increased air flow during this period.

Understanding the underlying causes of these distinct patterns in air flow is crucial for optimizing the HVAC system's performance. By analysing and interpreting these patterns, valuable insights can be obtained to guide further improvements and adjustments to ensure efficient and effective air flow within the system.

Based on these observations, the behaviours of the sensors appear to align with rational expectations in an HVAC system.

This relationship can be explained by the principles of fluid dynamics. As the fan or filter pressures decrease, it indicates lower resistance to airflow within the system. With reduced resistance, the air can flow more freely, resulting in an increase in the air flow rate. Conversely, when the fan or filter pressures increase, it suggests higher resistance, which can restrict the airflow and lead to a decrease in the air flow rate.

3.3. Data Visualization

Data visualization is a powerful tool for unsupervised data analysis. It helps researchers to identify patterns, trends, and relationships within complex data sets, as well as communicate those findings to others in a way that is easy to understand.

Data visualization is the process of representing data through visual elements, such as charts and graphs, to facilitate understanding and analysis. According to (Kirk, 2012). data visualization helps to overcome the limitations of traditional tabular or textual data representations by enabling humans to process visual information more quickly and easily.

In unsupervised data analysis, visualizations are particularly important for identifying patterns and trends that may be difficult to discern from raw data alone. In (Sacc{\`a}, 2019), it is explained that data visualization can reveal underlying structures and relationships within a dataset that may be obscured by its complexity. By using visualizations to identify clusters, outliers, and other patterns, researchers can gain insights that may not be apparent through traditional analysis methods.

In addition to aiding analysis, data visualization can also communicate the findings of unsupervised data analysis to others in a clear and concise manner. According to (Telea, 2014), visualizations are often more effective than textual or numerical summaries because they allow audiences to see the patterns and relationships for themselves.

3.4. The Analysis Techniques

3.4.1. Correlation Matrix and Correlation Heatmap

Both the Correlation Matrix and Correlation Heatmap are data visualizations commonly used in data analysis. According to (Mastrandrea, 2022), a Correlation Matrix is a table that displays the correlation coefficients between variables in a dataset. The correlation coefficient is a statistical measure of the strength of the relationship between two variables. In a correlation matrix, the values in the table represent the strength and direction of the correlation between each pair of variables. Correlation matrices are often used in exploratory data analysis to understand the relationships between variables.

A Correlation Heatmap is a graphical representation of a correlation matrix, in which the correlation coefficients are color-coded to represent their strength. (Ellison, 1994) explains that the heatmap displays the correlations between each pair of variables in a dataset as a grid of squares, with the color of each square indicating the strength and direction of the correlation. Correlation heatmaps are particularly useful for visualizing large correlation matrices and identifying patterns of relationships between variables.

Both Correlation Matrix and Correlation Heatmap are important tools in data analysis, as they allow for the quick identification of strong relationships between variables and can provide insights into how different variables are related to each other. According to (Kumar, 2022), these visualizations are commonly used in fields such as finance, economics, and social sciences to explore relationships between variables and to develop predictive models.

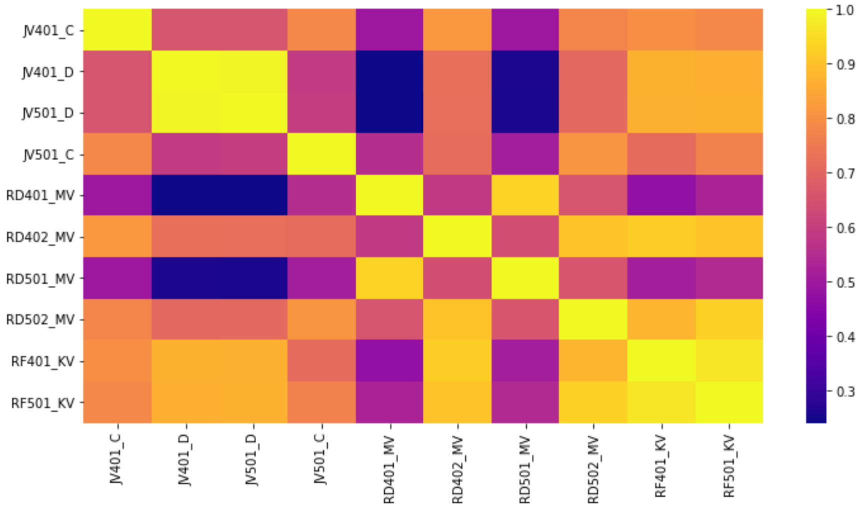


Figure 24: The Heatmap

In correlation heatmap the light color shows the high correlation between the variables and dark color shows the low correlation between the variables. Then we can see in this heatmap that all variables correlated with each other, and in the result, we can use the Principal Component Analysis (PCA) for dimensional reduction, because if there was no correlation between variables, we don't need to do PCA analysis (no correlations mean all dimensions are orthogonal).

3.4.2. The Pairs Plot

A pairs plot, also known as a scatterplot matrix, is a type of data visualization used to explore the relationships between multiple variables (Emerson, 2013). It allows for the simultaneous display of pairwise relationships between variables in a dataset, making it useful for identifying patterns and trends that may not be apparent from examining individual variables. Pairs plots can also be used to identify the most effective combination of features for describing the correlation between two variables or to distinguish distinct clusters.

In a pairs plot, each variable in the dataset is plotted against every other variable. The resulting scatterplots can be used to visualize the correlation between each pair of variables, as well as any non-linear relationships. Additionally, by plotting lines or creating linear separations, the pairs plot can aid in creating straightforward classification models.

Pairs plots are commonly used in fields such as machine learning, data analysis, and statistical modelling (McKinney, 2010). They can be created using various software packages, including R, Python, and MATLAB (MathWorks, 2021) (Patil, 2021).

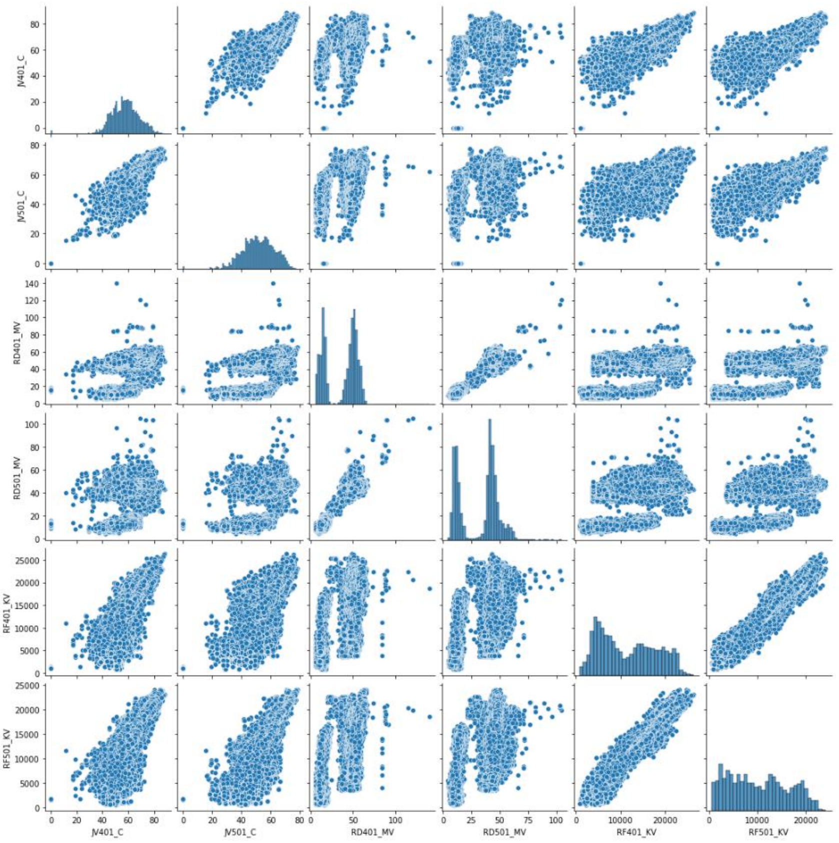


Figure 25: The Pairs Plot

We can see that some variables have a linear combination between each other like (JV401_C, JV 501_C, RF401_KV and RF501_KV), and other variables have nonlinear combination between each other like (RF401_KV, RD401_MV and RD501_MV), and so on.

3.4.3. Principal Component Analysis (PCA) Plot

A Principal Component Analysis (PCA) plot is a type of data visualization used to reduce the dimensionality of data in unsupervised data analysis (Jolliffe, Principal component analysis for special types of data, 2002). It shows the relationships between the principal components of the data and can help to identify clusters or patterns in the data.

PCA is a mathematical technique used to transform high-dimensional data into a lower-dimensional space while retaining most of the variation in the original data (Abdi, 2010). The resulting principal components are orthogonal to each other and represent different sources of variation in the data. By plotting the data in the space defined by the first two or three principal components, a PCA plot can show the relationships between the variables in a dataset and reveal any underlying patterns or structures.

PCA plots are commonly used in fields such as biology, chemistry, and finance to visualize high-dimensional datasets and identify patterns or clusters (Martens, 2001) (Wold, 1987).They can be created using various software packages, including R, Python, and MATLAB (MathWorks, 2021) (Patil, 2021).

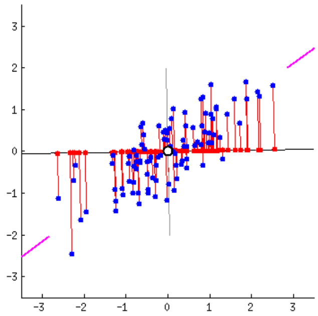


Figure 26: Dimensionality Reduction and Visualization Using PCA (Principal Component Analysis (PCA) Explained | Built In)

3.4.4. Density Plot

Density plots are a data visualization technique that display the probability density of a variable. They are similar to histograms, but instead of showing the frequency distribution of a variable, they show the probability density of the variable. Density plots can be useful in unsupervised data analysis to identify clusters or patterns in the data (Wilke, 2019) (Wickham, 2016).

3.5. Data Pre-Processing

Data pre-processing is an important step in the predictive maintenance and HVAC system analysis as it helps to clean, transform, and prepare the data for further analysis. The purpose of data pre-processing is to ensure that the data is in a consistent and usable format for further analysis and modelling. This can help to improve the accuracy and reliability of predictive models and reduce the risk of errors in the analysis.

There are several reasons why data pre-processing is necessary in predictive maintenance and HVAC system analysis:

- Handling missing values: Data collected from HVAC systems can often be incomplete or contain missing values, which can impact the accuracy of predictive models. Data pre-processing can help to identify and fill in missing values (Joshi, 2023).
- Data normalization: Predictive models are often sensitive to the scale of the input data, so it's important to normalize the data to ensure that it's in a consistent format. Data pre-processing can help to normalize the data by transforming it into a standard scale (Esteves, 2022).
- Removing irrelevant or redundant data: Predictive maintenance and HVAC system analysis can often involve large datasets with a lot of irrelevant or redundant information. Data pre-processing can help to identify and remove irrelevant or redundant data, reducing the risk of overfitting and improving the performance of predictive models (Aamo, 2018).
- Handling outliers: Outliers can have a significant impact on the accuracy of predictive models, so it's important to identify and handle them. Data pre-processing can help to identify and remove outliers, improving the reliability of predictive models (Joshi, 2023).

The data reprocessing in this study involves the following steps:

- Reading in multiple data files from a directory and concatenating them into a single pandas dataframe.
- Resampling the data to hourly intervals and calculating the mean of each interval.
- Filling in missing values using forward fill and dropping any remaining missing values.
- Standardizing the data using scikit-learn's StandardScaler function.
- Applying PCA to reduce the dimensionality of the data while preserving important information. This simplifies the data and improves the efficiency and accuracy of subsequent analysis.

- Apply the density-based clustering algorithm DBSCAN to the reduced dataset, this algorithm assigns each data point to a cluster or identifies it as noise.

3.5.1. Data cleaning

following data cleaning steps were taken:

- Combining data from multiple sources and aggregating them into a single dataset for easier analysis.
- Changing the frequency of the time series data to make it more manageable.
- Checking for missing data, abnormal values, or patterns that do not make sense to ensure data is reasonable.
- Filling in any missing data with an appropriate value to avoid problems in analysis.
- Converting timestamps to an index to facilitate time series analysis.

3.5.2. Data transformation

First, the data is loaded from pickle files and resampled to hourly intervals. The missing values are filled using forward-fill method and then scaled using StandardScaler.

Next, PCA is applied to the scaled data to reduce its dimensionality while retaining most of its variance. The number of components is chosen to capture 95% of the variance in the data.

Finally, the reduced data is clustered using DBSCAN algorithm, which generates labels for each data point indicating its cluster membership.

In summary, the data transformation involves reducing the dimensionality of the data using PCA and clustering the reduced data using DBSCAN algorithm.

3.6. Model Development

- The model development was performed using Python and several libraries such as Pandas, NumPy, and scikit-learn. The cleaned and reprocessed data was used to develop a machine learning model for predicting the risk of failure in the HVAC system.
- To develop the model, we started with a dataset of 10 features. However, due to the high dimensionality of the data, we applied Principal Component Analysis (PCA) to reduce the number of features, while still retaining most of the variability in the original data.
- Next, we applied Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to cluster the data into groups based on their density. we found that the data could be divided into four distinct clusters, with some additional data points classified as noise.
- We evaluated the effectiveness of the clustering using various performance metrics, such as Cross-validation scores, Silhouette score, and Homogeneity score. Additionally, we visualized the clusters using scatter plots and heatmaps to gain insight into the underlying structure of the data.
- Then, we used the clusters to develop a predictive model that can classify new data points based on their similarity to the existing clusters. we used a supervised learning algorithm, logistic regression, to build the model and evaluated its performance using cross-validation techniques.
- Finally, we repeated the last step by applying K-means clustering algorithm instead of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. Then we compared the results obtained from each cluster algorithm to determine which algorithm was the most effective for my data.
- Finally, we applied the decision tree regression to build the prediction model to predict the filter pressure, whereby monitoring this pressure, can we get insight about the filter situation.

3.6.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of data by transforming it into a new coordinate system called principal components (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009). The first principal component captures the maximum amount of variation in the data, while each subsequent component captures as much remaining variation as possible (Jolliffe, Principal component analysis for special types of data, 2002). PCA has several applications, including data visualization, feature extraction, and data compression (Abdi, 2010).

The intuition behind PCA is that high-dimensional data can be difficult to visualize and analyze. By reducing the number of dimensions, we can simplify the data and gain insights that would be difficult to obtain otherwise. PCA works by finding a linear combination of the original variables that captures as much variation as possible. This new variable is called the first principal component. Each subsequent component is orthogonal to the previous ones and captures as much of the remaining variation as possible (Jolliffe, Principal component analysis for special types of data, 2002).

PCA can be expressed mathematically as follows:

1- Given a data matrix X with n observations and p variables, we first standardize the data by subtracting the mean and dividing by the standard deviation for each variable:

$X' = (X - \mu) / \sigma$

where μ is the mean vector and σ is the standard deviation vector for each variable (Pearson, 1901).

2- Next, we calculate the covariance matrix S:

$S = (1/n) * X'X$

where n is the number of observations (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009).

3- We then find the eigenvectors and eigenvalues of S:

$S\ v = \lambda\ v$

where λ is the eigenvalue, and v is the eigenvector. Eigenvectors are unit vectors that describe the directions in which the data varies the most, while eigenvalues indicate the amount of variation in each direction (Jolliffe, Principal component analysis for special types of data, 2002) (Abdi, 2010).

4- We sort the eigenvectors in decreasing order of their eigenvalues, and select the top k eigenvectors to form a new matrix W. The number of components k can be chosen based on a criterion such as the percentage of variance explained or the elbow method (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009) (Abdi, 2010).

5- Finally, we transform the original data into the new coordinate system by multiplying it by W:

$$Y = X'W$$

The resulting matrix Y has the same number of observations as X, but with k variables instead of p (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009).

The loadings of each principal component can be used to interpret the relationships between variables in the original dataset. Positive loadings indicate variables that are positively correlated with the principal component, while negative loadings indicate variables that are negatively correlated. The first few principal components often correspond to the most important features of the dataset and can be used for data visualization or further analysis (Jolliffe, Principal component analysis for special types of data, 2002).

PCA has several assumptions, including the assumption of linearity and the sensitivity to outliers. Additionally, PCA may not be appropriate for all types of data, such as categorical data or data with missing values. However, when applied appropriately, PCA can be a powerful tool for dimensionality reduction and data analysis (Hastie T. a., The elements of statistical learning: data mining, inference, and prediction, 2009) (Jolliffe, Principal component analysis for special types of data, 2002) (Abdi, 2010).

3.6.2. Clustering Algorithms

Clustering is an unsupervised learning technique that aims to group together similar data points based on their features. In this study, we applied two commonly used clustering algorithms: DBSCAN and K-means.

3.6.2.1. DBSCA Clustering Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups together data points that are close to each other in a high-density region and separates out data points that are located in low-density regions. DBSCAN requires two input parameters: (eps) which specifies the radius of the neighbourhood around each point, and (min_sample) which specifies the minimum number of points in a neighbourhood to form a dense region (cluster).

The algorithm proceeds as follows:

1. For each data point, calculate the distance to all other data points (Ester, 1996).
2. If the distance is less than (eps(ε)), add the data point to the neighbourhood of the current point.
3. If the size of the neighbourhood is greater than or equal to MinPts (min_sample), the current point is marked as a core point (Ester, 1996).
4. Expand the cluster by adding all directly reachable points to the cluster. A point is directly reachable from another point if it is within (eps) distance.
5. If the point is not a core point but is within (eps) distance of a core point, it is added to the cluster (Ozgode Yigin, 2023).
6. Points that are not reachable from any core point are marked as noise (Ester, 1996).

DBSCAN has several advantages over other clustering algorithms, such as its ability to handle arbitrary shapes and its robustness to noise. However, it also has some limitations, such as its sensitivity to the choice of parameters eps and MinPts and its performance on datasets with different densities.

The equations used in DBSCAN are:

1. Euclidean Distance: Euclidean distance is used to measure the distance between two data points in DBSCAN. The formula for Euclidean distance is:

$$d(p, q) = \sqrt{\left(\sum(q_i-p_i)^2\right)}$$

where p and q are two data points, and i is the index of the feature (Ester, 1996).

2. Neighbourhood: The neighbourhood of a data point p is defined as all the data points that are within a distance of ε from p. The formula for the neighborhood is (Ester, 1996):

$N(p) = \{q \in D \mid dist(p,q) \leq \varepsilon\}$

where D is the dataset, q is a data point, and dist(p,q) is the distance between p and q.

3. Core Points: A core point is a data point that has at least MinPts other data points within its neighbourhood. The formula for core points is (Ester, 1996):

$|N(p)| \geq MinPts$

3.6.2.2. K-means Clustering Algorithm

The K-means clustering algorithm is a widely used unsupervised learning method for grouping similar data points together. The algorithm aims to partition a given dataset into K clusters, where K is a predetermined number of clusters chosen by the user.

The K-means algorithm works by iteratively assigning each data point to its nearest cluster center and updating the cluster centers based on the newly assigned data points. The objective function of the K-means algorithm is to minimize the sum of squared distances between each data point and its assigned cluster center.

The algorithm can be summarized in the following steps:

1. Initialize K cluster centers randomly in the feature space (developers, Clustering, 2007-2023).
2. Assign each data point to the nearest cluster center based on Euclidean distance (Luke, 2022).
3. Update the cluster centers as the mean of all data points assigned to that cluster (Sharma, 2019).
4. Repeat steps 2 and 3 until convergence (i.e., until the cluster assignments do not change).

The K-means algorithm can be formalized with the following equations:

- Let X = {x_1, x_2, ..., x_n} be the set of n data points, each with d dimensions.
- Let C = {c_1, c_2, ..., c_k} be the set of k cluster centers, each with d dimensions.
- Let S = {s_1, s_2, ..., s_n} be the set of n cluster assignments, where s_i is the index of the cluster to which x_i belongs.

Then, the K-means algorithm can be written as:

1. Initialize C randomly in the feature space (developers, Clustering, 2007-2023).
2. Repeat until convergence:
    a. For each data point x_i, assign s_i = argmin_j ||x_i - c_j||^2 (Luke, 2022).
    b. For each cluster center c_j, update c_j = mean({x_i | s_i = j}) (Sharma, 2019).

The K-means algorithm has some limitations, such as the sensitivity to the initial cluster center locations and the requirement of the number of clusters to be pre-specified. However, it remains a popular and effective clustering algorithm in many applications (developers, Clustering, 2007-2023) (Luke, 2022) (Sharma, 2019).

3.7. Evaluation Methods

In order to ensure the effectiveness of the PCA, DBSCAN clustering algorithms and K-means clustering algorithm, it is necessary to evaluate their performance. In this study, several evaluation methods are used to determine the optimal number of principal components and the number of clusters.

3.7.1. Scree Plot

The scree plot is a commonly used tool in PCA and clustering to determine the number of components or clusters to retain based on the amount of variance they explain. For PCA, the scree plot is a graphical representation of the eigenvalues of the principal components, with the eigenvalues on the y-axis and the number of principal components on the x-axis. The point where the plot starts to level off indicates the optimal number of components to retain (Aiyi Liu, 2002).

For clustering, a scree plot is a graphical representation of the eigenvalues or the sum of squared distances of a given set of data, sorted in descending order. In the case of K-means clustering, the scree plot is used to visualize the amount of variance explained by each cluster as a function of the number of clusters. The optimal number of clusters is typically chosen as the "elbow point" of the scree plot, which is the point where the decrease in variance explained starts to level off (Neeraj, 2020).

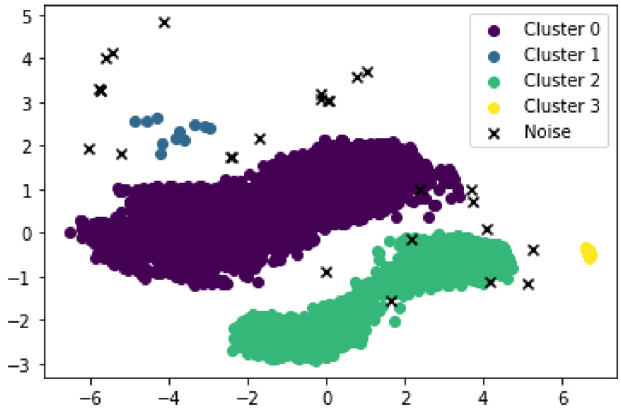


Figure 27: Clusters Generated by DBSCAN Algorithm

3.7.2. Elbow Method

The elbow method is a common method used to determine the optimal number of clusters in DBSCAN. The elbow method plots the average distance between each point and its k-nearest neighbors against the number of clusters. The point where the plot starts to level off indicates the optimal number of clusters (DAVID J. KETCHEN, 1996) (Neeraj, 2020).

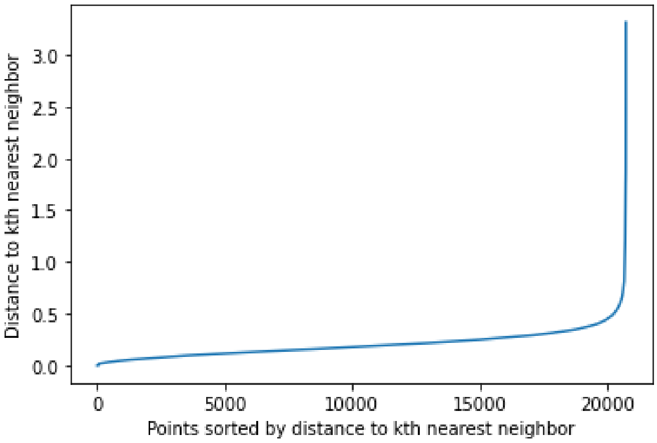


Figure 28: The Elbow Method

3.7.3. Silhouette Score

The silhouette score is a commonly used metric to evaluate the performance of clustering algorithms. It measures how similar a data point is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a score closes to 1 indicates a well-clustered data point, and a score close to -1 indicates a misclassified data point. The optimal number of clusters is typically chosen based on the highest average silhouette score (Rousseeuw, 1987).

3.7.4 Cross-Validation

Cross-validation is a technique used to evaluate the performance of a machine learning model. The process involves splitting the data into multiple subsets, where one subset is used as the test data and the other subsets are used for training the model. This process is repeated multiple times, with different subsets used for testing and training each time. The results of each iteration are then averaged to provide an overall performance metric for the model (Andreas C. Müller, 2017) (John D. Kelleher, 2018).

One commonly used metric for cross-validation is the cross-validation score. The cross-validation score is a measure of how well the model is able to generalize to new data. The score is calculated by taking the average of the scores obtained in each iteration of the cross-validation process (Tauzin, 2021) (Richie-Halford A, 2022).

The most commonly used form of cross-validation is k-fold cross-validation, where the data is divided into k subsets of equal size. The model is then trained on k-1 of these subsets and tested on the remaining subset. This process is repeated k times, with each subset used as the test data once (Kohavi, 1995).

3.7.5. Homogeneity Score

Homogeneity score is a measure used to evaluate the quality of clustering results. It measures the extent to which each cluster contains only members of a single class. In other words, a clustering result is considered to be homogeneous if all members of a given class belong to the same cluster (Meilundefined, 2005).

Homogeneity score can be calculated using different methods, such as the V-measure or the Adjusted Mutual Information (AMI) score. The V-measure is based on the harmonic mean of the cluster purity and the class purity, while the AMI score is a normalized mutual information measure that adjusts for chance (developers, sklearn.metrics.homogeneity_score, 2007-2023).

Homogeneity score is a useful evaluation metric for clustering algorithms, as it provides insight into the extent to which the clustering results reflect the true structure of the data. However, it should be used in combination with other evaluation methods to obtain a more complete understanding of the clustering performance (Jayaprakash, Nagarajan, Prado, Subramanian, & Divakarachari, 2021).

3.8. Decision Tree Regression

Decision Trees are a type of supervised machine learning algorithm used for both regression and classification tasks (Hastie et al., 2009). They are constructed by recursively partitioning the feature space into subsets, with each partition represented by a tree node. The goal of decision tree regression is to build a model that can predict a continuous target variable based on a set of input features.

3.8.1. Decision Tree Regression Algorithm

The decision tree regression algorithm works by recursively partitioning the feature space into subsets based on the input features, and fitting a simple regression model (e.g., mean or median) to each subset (Breiman, 1984). The algorithm follows a top-down approach, starting from the root node of the tree and splitting the data into two or more subsets based on the value of a single input feature. The splitting criterion is usually based on minimizing the variance or mean squared error of the target variable within each subset (Linoff, 2011). The decision tree regression algorithm can be summarized in the following steps:

- Choose the feature that best splits the data based on a chosen criterion (e.g., variance reduction, information gain) (Trevor Hastie, 2001).
- Split the data into two or more subsets based on the chosen feature value (Breiman, 1984).
- Repeat steps 1-2 for each subset until a stopping criterion is met (e.g., maximum tree depth, minimum number of samples per leaf node) (avcontentteam, 2016).

- Fit a simple regression model to each leaf node of the tree (e.g., mean, median) (Trevor Hastie, 2001).

3.8.2. Advantages of Decision Tree Regression

One of the main advantages of decision tree regression is that it can handle both categorical and continuous input features, as well as mixed data types (Trevor Hastie, 2001). It is also a non-parametric algorithm, which means it makes no assumptions about the underlying distribution of the target variable. In addition, decision trees can be easily visualized and interpreted, which can help with understanding the model's predictions (Linoff, 2011).

3.8.3. Disadvantages of Decision Tree Regression

One of the main disadvantages of decision tree regression is that it can easily overfit to the training data, especially if the tree is allowed to grow to a large depth (Trevor Hastie, 2001). This can lead to poor generalization performance on new data. To avoid overfitting, pruning techniques can be used to reduce the size of the tree or restrict the complexity of the splitting rules (Trevor Hastie, 2001). Another disadvantage of decision tree regression is that it may not capture complex interactions between the input features, especially if the interactions are non-linear (Hastie T. , 2020).

3.9. Evaluate the Prediction Model

3.9.1. Mean Squared Error (MSE):

The Mean Squared Error (MSE) is a commonly used method to evaluate the performance of a regression model. It measures the average of the squared differences between the predicted and actual values.

The formula for MSE is (James, 2013):

$MSE = 1/n * \Sigma(y_i - \bar{y})^2$

where:

- n is the number of data points.

- y_i is the predicted value for the i-th data point.
- $\bar{y}$ is the actual value for the i-th data point.

3.9.2. Mean Absolute Error (MAE):

The Mean Absolute Error (MAE) is another commonly used method for evaluating regression models. Unlike the MSE, the MAE measures the average absolute difference between the predicted and actual values (James, 2013).

The formula for MAE is:

$MAE = 1/n * \Sigma|y_i - \bar{y}|$

where:

- n is the number of data points.
- y_i is the predicted value for the i-th data point.
- $\bar{y}$ is the actual value for the i-th data point.

3.9.3. R-squared (R2):

R-squared (R2) is a commonly used method for evaluating the goodness of fit of a regression model. It measures the proportion of the variance in the dependent variable that is explained by the independent variables (James, 2013).

The formula for R2 is:

$R^2 = 1 - (SS_\gamma - SS_i)/SS_\gamma$

where:

- SS_γ is the total sum of squares, or the sum of the squared differences between the actual values and the mean of the dependent variable.
- SS_i is the sum of squares of the residuals, or the sum of the squared differences between the actual values and the predicted values.

3.9.4. Coefficient of Determination (r):

The Coefficient of Determination (r) is also known as the Pearson correlation coefficient and measures the strength and direction of the linear relationship between two variables. It ranges from -1 to 1, where a value of 1 indicates a perfect positive correlation, 0 indicates no correlation, and -1 indicates a perfect negative correlation (Jones, 2015) (Mendenhall, 2012).

The formula for r is:

$r = \Sigma((x_i - \ddot{x})(y_i - \bar{y})) / \mathrm{sqrt}(\Sigma(x_i - \ddot{x})^2 * \Sigma(y_i - \bar{y})^2)$

where:

- x_i is the i-th value of the independent variable
- $\ddot{x}$ is the mean of the independent variable
- y_i is the i-th value of the dependent variable
- $\bar{y}$ is the mean of the dependent variable

3.10. Deployment

The deployment phase involves deploying the model in a real-world setting and monitoring its performance. The deployment phase is critical to assess the effectiveness of the model in a real-world scenario. However, due to the unavailability of sufficient data and observations, the deployment step has not been carried out yet. The temperature sensor, which is a critical component of the HVAC system, did not provide any readings. Instead, we collected data on pressure through the fans, pressure through the filters, air flows, and the percentage of opening the valve of fans. Moreover, we did not have enough observation periods, and the data was collected during the corona time, which may have affected the results. Therefore, future research should focus on gathering more data and observations and get observations from the temperature sensor in addition to the other sensors which I got my dataset from, to deploy and evaluate the model's real-life performance.

4. The Results

4.1. Understanding the Data

The following is a list of descriptions of which sensors are involved in this thesis, how to interpret the data from them, and how to relate this information to the problem of predictive maintenance of a HVAC system.

- JV401/501_C is a vane sensor used to measure the opening angle of a damper or valve in an HVAC system. The sensor consists of a small blade or vane that is attached to the damper or valve and can rotate as the damper or valve opens or closes. The vane is connected to a sensor that measures the angle of the vane relative to its fully open position and converts this angle into a percentage value. The percentage value obtained from JV401_C is typically used to monitor and control the airflow rate in an HVAC system. By adjusting the position of the damper or valve based on the JV401_C reading, the airflow rate can be regulated to meet the desired setpoint.
  JV401_C is an important component in an HVAC system, as it allows for accurate control of the airflow rate and helps to maintain comfortable and healthy indoor environments.

- JV401/501_D is a sensor used to detect the on/off status of a fan in the HVAC system.
  The sensor works by detecting the electrical signal that is sent to the device to turn it on or off. When the device is turned on, the sensor will register a signal and indicate that the device is in an "on" state. When the device is turned off, the sensor will no longer detect the signal and will indicate that the device is in an "off" state.
  The status information provided by JV402_D can be used for various purposes in an HVAC system, such as monitoring the operating status of equipment, troubleshooting faults, or triggering automated control sequences. By knowing whether a device is on or off, the system can make adjustments to maintain the desired performance and efficiency levels.

- RD401/501_MV are pressure sensors used to measure the pressure drop across the filters in an HVAC system. The sensors work by detecting the difference in pressure between the upstream and downstream sides of the filter.

When a filter is clean, there is little resistance to airflow, and the pressure drop across the filter is relatively low. As the filter becomes dirty and clogged with particulate matter, the resistance to airflow increases, and the pressure drop across the filter also increases. RD401/501_MV can measure this pressure drop and provide feedback to the HVAC system's control system.

The pressure drop information provided by RD401/501_MV can be used for various purposes in an HVAC system, such as monitoring filter performance, indicating when a filter needs to be replaced or cleaned, or triggering automated control sequences to maintain optimal filter efficiency.

By monitoring the pressure drop across the filters, HVAC systems can ensure that the filters are working properly and maintain healthy indoor air quality levels.

- RD402/502_MV are pressure sensors used to measure the pressure difference across a fan in an HVAC system. The sensors work by detecting the difference in pressure between the inlet and outlet sides of the fan.

  When the fan is running, it generates a pressure difference across its blades, which is necessary to move the air through the system. RD402/502_MV can measure this pressure difference and provide feedback to the HVAC system's control system.

  The pressure difference information provided by RD402/502_MV can be used for various purposes in an HVAC system, such as monitoring fan performance, indicating when a fan is not working properly, or triggering automated control sequences to maintain optimal fan efficiency.

  By monitoring the pressure difference across the fans, HVAC systems can ensure that the fans are working properly and maintain optimal airflow rates throughout the system. This is important for maintaining comfortable and healthy indoor environments, as well as for ensuring the efficient operation of the HVAC system.

- RF401/501_KV is a value that is obtained by multiplying the pressure difference across a fan by the k-factor, which is a mathematical constant that is specific to the fan. The k-factor is related to the geometry of the fan and is used to calculate the airflow rate through the fan based on the pressure difference across it. RF401/501_KV does not have a direct relationship with the system's filters. However, the performance of the filters can indirectly impact the pressure difference across the fan and therefore the RF401/501_KV value.

As the filters become clogged with dirt and debris, they restrict the airflow through the HVAC system. This can cause the pressure difference across the fan to increase, which in turn can cause the RF401/501_KV value to increase. If the filters become too clogged, the increased pressure difference can cause the fan to work harder and consume more energy, which can impact the efficiency of the system.

Therefore, maintaining clean and properly functioning filters is important for optimizing the performance of the HVAC system and ensuring accurate RF401/501_KV readings.

The correlation matrix of the dataset composed by the variables above is the following: in the table (8).

Table 8: The Correlation Matrix

	JV401_C	JV401_D	JV501_C	JV501_D	RD401_MV	RD402_MV	RD502_MV	RD502_MV	RF401_KV	RF501_KV
JV401_C	1.0	0.672	0.759	0.673	0.505	0.835	0.504	0.792	0.814	0.802
JV401_C	0.672	1.0	0.596	0.992	0.24	0.721	0.254	0.702	0.87	0.866
JV401_C	0.759	0.596	1.0	0.604	0.559	0.718	0.517	0.82	0.716	0.776
JV401_C	0.673	0.992	0.604	1.0	0.237	0.719	0.25	0.706	0.868	0.87
RD401_MV	0.505	0.24	0.559	0.237	1.0	0.588	0.934	0.656	0.47	0.526
RD401_MV	0.835	0.721	0.718	0.719	0.588	1.0	0.642	0.901	0.922	0.903
RD401_MV	0.504	0.254	0.517	0.25	0.934	0.642	1.0	0.656	0.509	0.541
RD401_MV	0.791	0.702	0.82	0.706	0.656	0.901	0.656	1.0	0.872	0.927

RF401_KV	0.814	0.87	0.716	0.868	0.47	0.922	0.509	0.872	1.0	0.964
RF401_KV	0.802	0.866	0.776	0.87	0.526	0.903	0.541	0.927	0.964	1.0

The matrix is by construction symmetric, with each element representing the similarity score between two items. The values range between 0 and 1, where 1 represents maximum similarity and 0 represents no similarity. where higher similarity indicates that the items share more features in common.

In other words, higher similarity indicates that users who extract information from one item are more likely to extract the same information from the other item as well.

- For example, JV401_C (The percentage of the fan valve in the return duct) has a high positive correlation with RD402_MV (the fan pressure in the supply duct)and RF401_KV (the air flow in the supply duct), and that means a strong positive relationship between the opening percentage of the fan valve and two other variables: the fan pressure in the supply duct and the air flow of the fan in the supply duct. Specifically, a high positive correlation of greater than 0.8 exists between these variables.

  So, as the opening percentage of the fan valve increases, both the fan pressure and air flow in the supply duct also increase in a linear fashion.

  And that is reasonable in HVAC, for example, if the opening percentage of the fan valve increases, this will allow more air to flow through the HVAC system, resulting in an increase in both the fan pressure in the supply duct and the air flow of the fan in the supply duct. Conversely, if the opening percentage of the fan valve decreases, this will decrease the amount of air flowing through the HVAC system, resulting in a decrease in both the fan pressure and air flow in the supply duct.

- The same for JV401/501_D (the on/off status of the fan) and the RF401/501_KV (the air flow), and that is reasonable to expect that the on/status of the HVAC system would have a positive correlation with the air flow in the system. Specifically, when the HVAC system is turned on or in an active state, there should be a positive correlation with the air flow in the system.

When the HVAC system is turned on, the fan in the system should begin to circulate air throughout the system. This fan is responsible for drawing in air from the return ducts, pushing it through the filter and then into the supply ducts for distribution throughout the building. As a result, the air flow in the system should increase when the system is turned on or in an active state.

- The high positive correlation between the RD401/501_MV (the filter pressure), and the RD402/502_MV (the fan pressure), indicates that there is a strong relationship between these two variables. Specifically, as the filter pressure increases, there will be a corresponding increase in the fan pressure.

  This relationship is to be expected, as the filter pressure is a measure of the pressure drop across the air filter in the HVAC system. As the air filter becomes dirty or clogged, the pressure drop across the filter will increase, which will cause the fan to work harder to move air through the system. This increased effort by the fan will result in an increase in the fan pressure. By understanding the relationship between the filter pressure and the fan pressure, HVAC professionals can optimize the performance of the system and ensure that it is operating efficiently. For example, if the filter pressure is consistently high, it may be necessary to change the air filter or adjust the fan settings to compensate for the increased pressure drop. Similarly, if the fan pressure is consistently high, it may be necessary to inspect the system for blockages or other issues that are causing the fan to work harder than it should be.

Based on the results, it was observed that the correlation between RD401_MV (the filter pressure in the supply duct) and RD502_MV (the fan pressure in the return duct) was stronger than the correlation between RD501_MV (the filter pressure in the supply duct) and RD502_MV (the fan pressure in the supply duct). This suggests that there is a closer relationship between the filter and fan pressures when measured in different locations within the HVAC system. This is reasonable due to the fact that the system operates as a temperature regulator with a set point of 21.0°C. we can see in the diagram of the HVAC system, that the temperature in the supply duct (24.3°C) is higher than the temperature in the return duct (22.0°C). Therefore, the reference point in the system can be considered as the temperature in the return duct where the air flows out of the room. This point should have a temperature that is equal to or close to the setpoint temperature that should be the room temperature, indicating that the fan in the return duct works as a controller of this system. Improving the correlation between filter pressure and fan pressure in the supply duct could potentially involve adjusting the system design to ensure that the reference point is closer to the room temperature, such as by increasing the temperature in the supply duct or by relocating the reference point.

4.2. Reprocessing the Data

The following list describes which data reprocessing steps were taken to prepare the dataset for a further analysis. In this thesis we have been, using Python; this means that we use a lingo that refers to that specific programming language.

1. The raw data was stored across multiple files in a directory. To preprocess the data, the files were read in and concatenated into a single pandas dataframe using Python code. The code involved navigating to the root directory containing the data files and iterating through subdirectories for each HVAC system and year of data. For each data file, the code read in the file as a pandas dataframe, resampled the data to hourly intervals and calculated the mean of each interval, and added it to a list. The code then concatenated all dataframes in the list into a single dataframe. Missing values were filled in using forward fill and any remaining missing values were dropped, filling in the missing values improved the accuracy of the predictive models by ensuring that the data was complete and consistent. Without filling in missing values, the models could have been impacted by inaccurate or incomplete data. Therefore, this step was critical for ensuring the reliability and validity of subsequent analyses, then the column names were standardized and the resulting dataframe was resampled again to hourly intervals and averaged.

The resulting preprocessed data consists of (20738) rows and (10) features, as shown in the figure below.

Table 9: The Dataset Description Before Normalizing the Data

Timestamp	JV401_C	JV401_D	…	RF401_KV	RF501_KV
2019-05-28 07:00:00+00:00	64.950994	0.500000	…	21541.168724	21708.235670
2019-05-28 08:00:00+00:00	65.253974	0.500000	…	21911.909218	22248.202888
2019-05-28 09:00:00+00:00	65.253974	0.500000	…	21911.909218	22248.202888
2019-05-28 10:00:00+00:00	42.329308	0.333333	…	11499.278628	9755.888373
2019-05-28 11:00:00+00:00	73.741399	0.666667	…	21905.183605	22271.447558
…	…	…	…	…	…
2021-10-08 04:00:00+00:00	55.282519	0.928571	…	12488.583433	10898.649557
2021-10-08 05:00:00+00:00	59.128727	1.000000	…	14470.727463	12719.211227
2021-10-08 06:00:00+00:00	62.892149	0.857143	…	15477.629996	13370.591533
2021-10-08 07:00:00+00:00	63.096610	0.857143	…	14578.115768	13111.471836
2021-10-08 08:00:00+00:00	61.042864	1.000000	…	15718.233051	13881.189540

2. Next, the resulting dataframe was standardized using the StandardScaler function from the scikit-learn library in Python. This function scales the data to have a mean of 0 and a standard deviation of 1. The resulting scaled data was stored in a new pandas dataframe called df_scaled. The mean and standard deviation of the scaled data were calculated using the numpy library in Python and were found to be:

- The mean values of the variables are:

    [6.142e-16, 1.316e-16, 1.316e-16, -2.083e-16, 1.754e-16, -1.316e-16, -2.631e-16, -2.851e-16, -1.316e-16, 4.386e-17]

- The standard deviations of the variables are:
  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.].

3. After standardizing the dataset, PCA was applied to reduce the dimensionality of the data. PCA was performed on the dataset, resulting in four principal components explaining 95% of the variance (with 4 components).
4. The original dataset was then transformed into a reduced dataset with four principal components, which captured most of the variability in the data. I used Scikit-learn library for this step.
5. The DBSCAN clustering algorithm was applied to the PCA-reduced dataset in order to identify any underlying structure in the data. The resulting clustering identified 4 distinct clusters in the data, as well as some noise points that did not belong to any cluster.

The identification of clusters in the data suggests that there may be underlying patterns or relationships between the variables that can be further explored. For example, the features that are most strongly associated with each cluster can be identified by calculating the loadings of each variable on the principal components. This can provide insight into which variables are most important for predicting filter failure and can help guide the development of a predictive maintenance model, (that will be explained in the next sections).

4.3. Exploratory Data Analysis

4.3.1. Principal Component Analysis

Applying a PCA (Principal Component Analysis) analysis resulted in a reduced dataset expressed by four orthogonal components, instead of 10 original variables, that constitute approximately 95% of the total explained variance that expressed by the original dataset.

To visualize the results of PCA we proceed by plotting:

First, the 3D scatter plot of principal components (PCs): this plot shows the distribution of my data in a three-dimensional space, where the x-axis represents PC1, the y-axis represents PC2, and the z-axis represents PC3. PC4 is not represented in the plot as it is not one of the three dimensions shown.

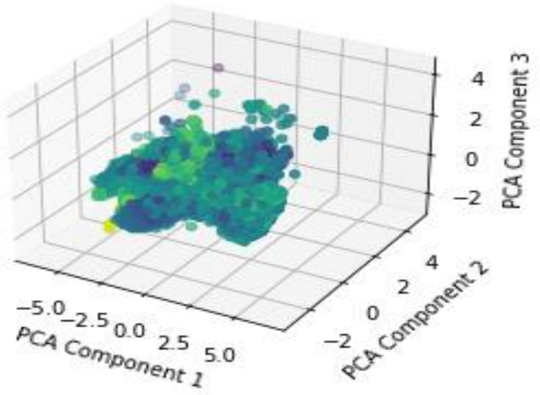


Figure 29: The 3D Scatter Plot of Principal Components

The explained variances for four components which are:

PCA explained-variance-ratio: [0.72829003 0.15441343 0.04834202 0.02745853]

The sum of these variances is: PCA explained-variance-ratio: 0.9585.

Then, we proceed by plotting the loadings of the variables in the dataset with two PCs: loadings represent the contribution of each original variable to each PC, so plotting the loadings for each pair of PCs can help to identify patterns or relationships between variables, the variables are represented as vectors.

1. Plot of the loadings with the first two components (PC1 and PC2), is shown in figure_1:

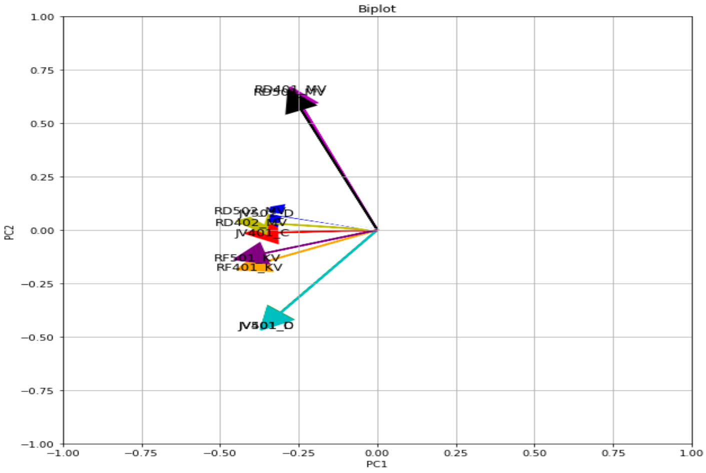


Figure 30: Plot of The Loadings with The First Two Components (PC1 and PC2)

2. Plot of the loadings with first and third components (PC1 and PC3), is shown in figure_2:

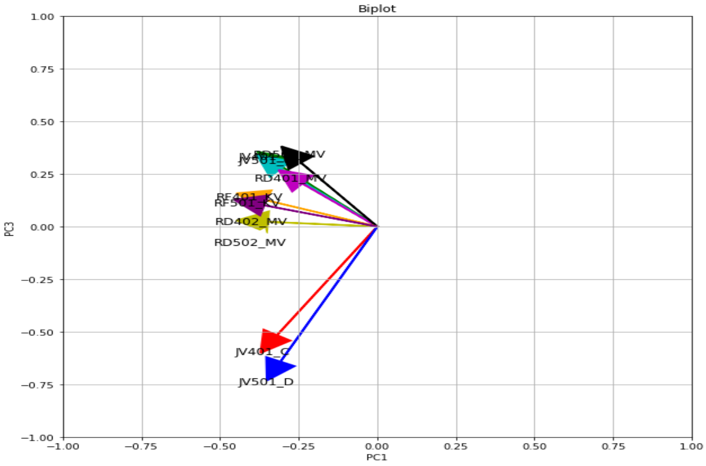


Figure 31: Plot of The Loadings with The First Two Components (PC1 and PC3)

3. Plot of the loadings with first and fourth components (PC1 and PC4), is shown in figure_3:

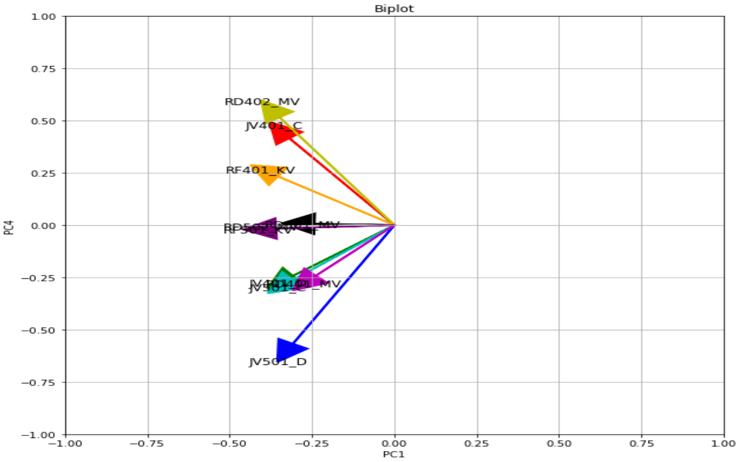


Figure 32: Plot of The Loadings with The First Two Components (PC1 and PC4)

The direction of a vector represents the direction in which the variable changes the most. The length of the vector represents the magnitude of the change.

The variables that have vectors that are close together in the plot, are highly correlated with each other like, (RD401/501_MV (the filter pressures) and RD402/502_MV (the fan pressures)), while variables that are uncorrelated, they have vectors that are orthogonal (perpendicular) to each other like (JV401_C (the percentage of the fan valve in the supply duct)) and JV501_D (the fan on/off status in the return duct)).

4.3.2. Evaluating the Results from the Principal Component Analysis

4.3.2.1. The Scree Plot

The scree plot is a graphical representation of the eigenvalues of the principal components in a principal component analysis (PCA). It is a simple line plot of the eigenvalues versus the component number, sorted in decreasing order.

The scree plot is used to visually assess the number of significant principal components in a dataset. The eigenvalues represent the amount of variance explained by each principal component, and a high eigenvalue indicates that the corresponding component explains a large amount of variance in the data. The scree plot shows the relative contribution of each principal component to the total variance in the dataset.

The scree plot is named after the steep slope ("scree") that is often seen in the plot at the point where the eigenvalues start to level off. This point represents the number of principal components that should be retained for further analysis, as additional components beyond this point are unlikely to contribute much to the total variance explained.

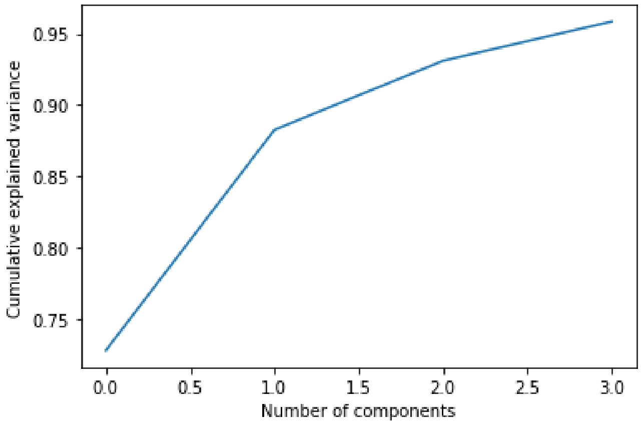


Figure 33: Visualization the Number of Significant Principal Components

We can see that the 3 components describe a 95% of the total variance that means my previous result is correct that 4 components explain almost 96% of the total variance.

4.3.3 Clustering Analysis

4.3.3.1. DBSCAN Clustering Results

We seek to find whether some data cluster together, for the primary purpose of verifying whether it is possible to find a 'normal operation mode' to be compared against a faulty one.

The approach to clustering involved in this thesis is via using DBSCAN on the reduced data obtained through PCA. The best values for the hyperparameters were determined by iterating over a range of values for the epsilon and minimum sample size parameters, and evaluating the resulting clusters using the silhouette score, homogeneity score, cross validation scores, and the model accuracy. The best values were found to be an epsilon value (eps=0.8) and a minimum sample size (min-samples=5).

To evaluate the optimal hyperparameters, I experimented the knee plot which is a graph of the sorted distances between each point and its k-th nearest neighbor. The choice of n_neighbors for the NearestNeighbors algorithm was taken as (n_neighbors=20) and this means that for each point, the code is computing the distances to its 20 nearest neighbors. Then I visualised inspect the plot to identify the knee, which represents the optimal value of epsilon.

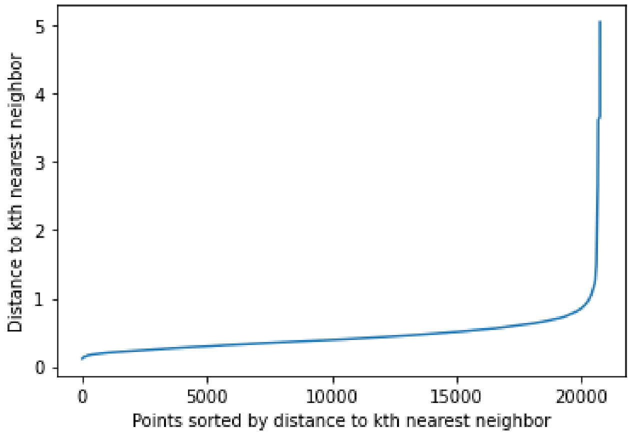


Figure 34: The Knee Plot to Evaluate the Optimal Hyperparameters

The knee plot shown that the optimal value of the hyperparameter 'eps' is at the knee or the elbow point where the rate of decrease in similarity value slows down significantly, and it is very close to the (0.8).

Then, we characterized the clusters based on their size and average distance between points within each cluster. The clusters were visualized using a scatter plot, with each cluster assigned a different color and symbol. The plot showed that the clusters were well-separated in the PCA space, with little overlap between them.

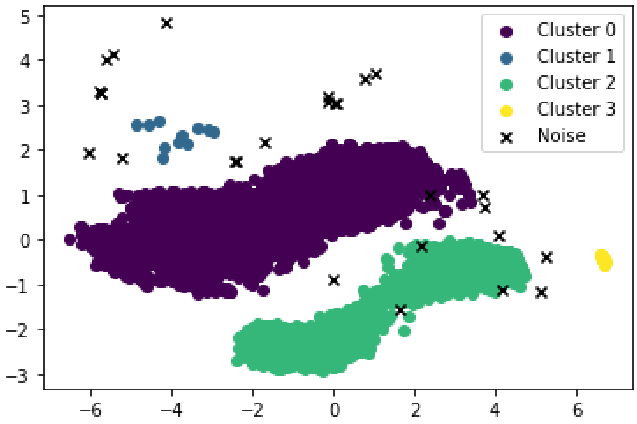


Figure 35: DBSCAN Clusters

The clustering analysis resulted in four clusters and some points were classified as noise (as shown in figure 37). The two large clusters account for the majority of the data points, while the two smaller clusters are relatively small in comparison. This suggests that there are two distinct regions in the data with high point density and separated by areas of lower point density.

Then to visualize the DBSCAN clustering results, we created a scatter plot of the data using the first two principal components on the x and y axes, and use different colors for each cluster, as the following figure:

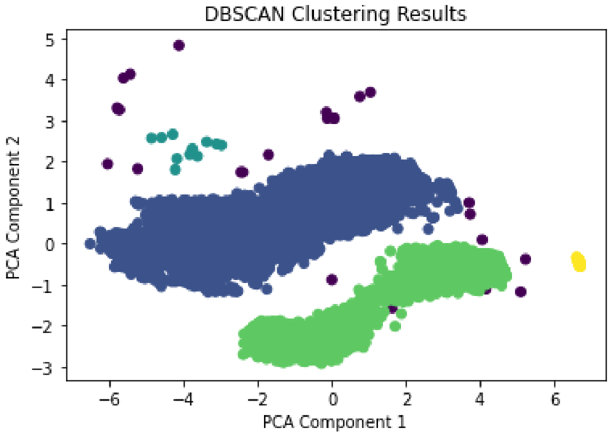


Figure 36: Visualize the DBSCAN Clustering Results with PCA

We obtained two large clusters and two very small clusters, in addition to some noise points that are not assigned to any cluster. The two large clusters corresponded to the two clouds of data points that were located from the top right downward gradient to the bottom left in the plot. The DBSCAN algorithm was able to identify the two large clusters corresponding to the two clouds of data points because they had high point density and were separated from other regions by areas of lower point density. The two smaller clusters correspond to smaller dense regions in the data, while the noise points did not satisfy the density criterion and they are not assigned to any cluster.

4.3.3.2. The PCA Results As a Biplot

We used PCA to reduce the dimensionality of my data, and obtained the first two principal components, which represent the two-dimensional space of my biplot.

The biplot is created by plotting the transformed observations (scores) on the two-dimensional space, and drawing lines from the origin to the point corresponding to each variable's loading. Loadings represent the correlations between the variables and the principal components. The length of each line represents the strength of the correlation, and the angle between two lines represents the angle between the two corresponding variables.

In addition, we used the DBSCAN clustering algorithm to group my observations into clusters based on their proximity in the transformed space. The resulting labels were then used to color-code the observations in the biplot.

The biplot function takes four arguments: the scores (the transformed observations df_scaled), the loadings, the list of variable names, and an optional list of integers to use as labels. The function then scales the scores, plots them as points, and draws lines from the origin to the loading points for each variable. Finally, the function displays the plot, with the observations colored according to their assigned cluster label. And the biplot result shown in figure:

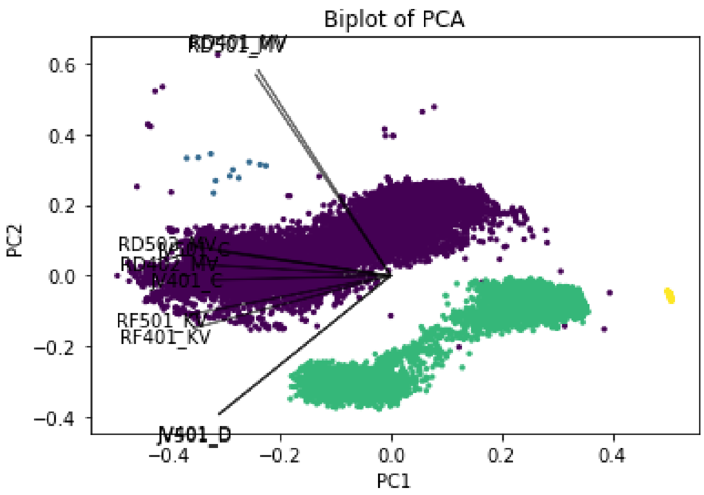


Figure 37: Biplot PCA

We can see from the previous plot that, all variables have negative loadings on the first principal component (PC1), while 5 variables have negative loadings and 5 variables have positive loadings on the second principal component (PC2). This means that the variables are strongly correlated with PC1 but in an opposite direction, while they are less strongly correlated with PC2 and some variables are positively correlated with it and some are negatively correlated.

The variables with strong negative loadings on PC1 are likely to be negatively correlated with each other, meaning that when one of these variables increases, the others tend to decrease. Similarly, the variables with strong positive loadings on PC2 are likely to be positively correlated with each other, meaning that when one of these variables increases, the others tend to increase as well. The loadings in each principal components are showen in the following table:

Table 10: The Loadings in Each Component

The variable	PC1	PC2	PC3	PC4
'JV401_C'	-0.318597	-0.012263	-0.515903	0.415335
'JV401_D'	-0.307925	-0.387711	0.288228	-0.242612
'JV501_C'	-0.307681	0.067457	-0.641694	-0.567270
'JV501_D'	-0.308188	-0.389609	0.273702	-0.261865
'RD401_MV'	-0.238069	0.577204	0.204253	-0.241884
'RD402_MV'	-0.348312	0.031948	0.022522	0.515000
'RD501_MV'	-0.242665	0.564571	0.303423	0.006011
'RD502_MV'	-0.350774	0.081118	-0.062570	-0.005564
'RF401_KV'	-0.353371	-0.147406	0.125271	0.232798
'RF501_KV'	-0.359491	-0.109317	0.100915	-0.016532

As we see that the loadings on PC1 indicate that the percentage of fan valve in the supply and return duct, the on/status of fan in the supply and return duct, the fan pressure, the air flow and the filter pressure in the supply duct are all strongly correlated with each other, meaning that they tend to vary together in the opposite direction, this is consistent with the idea that changes in the fan valves and fan status can affect the air flow and pressure in the ducts, which in turn can impact the efficiency and performance of the HVAC system.

On the other hand, the variables with the strongest positive loadings on PC2 are the percentage of the fan valve in the return duct, the filter pressure in the supply duct, the filter pressure in the return duct, and the fan pressure in the return duct. This suggests that these variables are positively correlated with each other, it suggests that they tend to vary together in the same direction. In this case, high values of filter pressure in the supply and return ducts and fan pressure in the return duct are all associated with high values of PC2. Similarly, low values of these variables are associated with low values of PC2.

The fact that these variables have positive loadings on PC2, but not as strong as the loadings on PC1, suggests that they are less important in explaining the overall variance in the data compared to the variables that have strong negative loadings on PC1. However, they still contribute to the overall structure of the data and may be relevant for understanding the operation of the HVAC system.

4.3.3.3. K-means Clustering Results

To determine a meaningful number of clusters from the data one can apply a K-means clustering algorithm to the reduced data then determine the number of clusters using a scree plot. Specifically, the first two principal components of my data were used and performed K-means clustering with a range of cluster numbers from 1 to 10. Then we generated a scree plot to identify the "elbow" of the curve, which indicates the optimal number of clusters. Based on the scree plot, we determined that the optimal number of clusters was choosed to be (5) clusters.

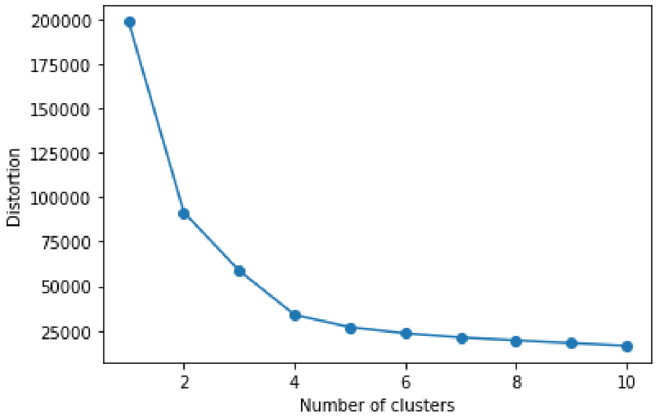


Figure 38: The Optimal Number of K-means Clusters Using a Scree Plot

Once the optimal K value (k =5) is determined, the K-means algorithm is applied to the data by initializing K random cluster centroids and iteratively assigning each data point to its closest centroid and updating the centroid location based on the new cluster assignments. This process is repeated until the algorithm converges and the cluster assignments remain constant. Then, we Performed k-means clustering with the chosen number of clusters.

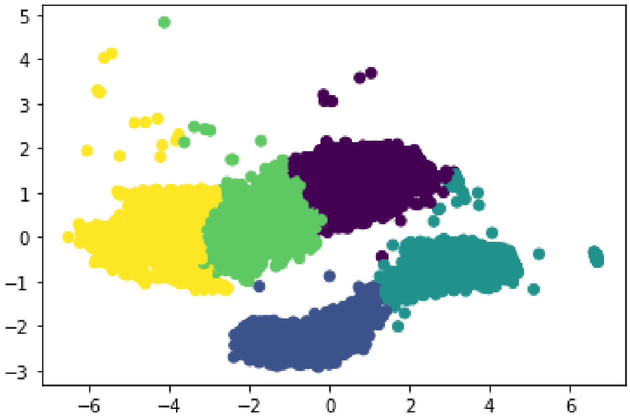


Figure 39: The Clusters Generated By K_means

After that, we visualized the clustered data using a scatter plot. The scatter plot shows the first two principal components on the x and y axes, with the data points color-coded based on their cluster labels. As shown in Figure 42, we can see that the K-means algorithm successfully separated the data into (5) distinct clusters.

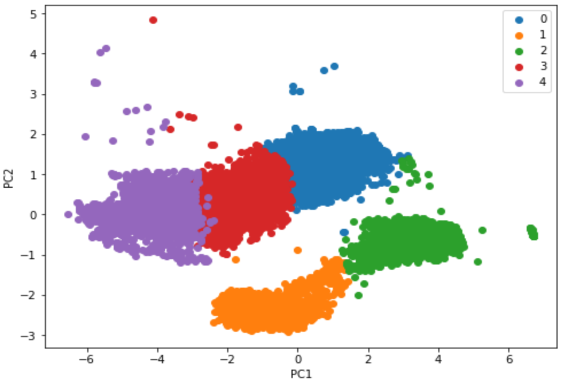


Figure 40: Visualize the K-means Clustering Results with PCA

We can see here that the K-Means was able to identify more spherical clusters, and that discribes why K-Means is more suitable for datasets with well-defined spherical clusters.

4.3.4. Comparing The Clustering Results

My data has two distinct clouds when we applied the DBSCAN clustering algorithm, that are arranged in a diagonal pattern from the top right to the bottom left, with one cloud larger than the other, and the upper one (cluster 0) has the observations that have values larger than those that are included in the lower cluster (cluster 2).

And when we applied k-means clustering with 5 clusters, the algorithm divided the data into 3 clusters in the larger cloud and 2 clusters in the smaller cloud, along with some separate points that were assigned to the different clusters, that may indicate that there are more subtle differences in the data within the larger cloud. So, we can say that there are two main factors that are driving the differences between the data points: one factor that is related to the performance characteristics of the larger cloud (such as the percentage of fan valve, fan pressure, filter pressure, air flow, and fan status), and another factor that is related to the performance characteristics of the smaller cloud. Within each cloud, there may be further variations in performance characteristics that are captured by the different clusters.

4.4. Machine Learning Results

After applying the DBSCAN clustering algorithm to the reduced dataset with four principal components, we performed further analysis on the clustered data using supervised machine learning techniques. Specifically, we applied logistic regression to the clustered data to predict the cluster membership of each data point. we used Scikit-learn library to perform this step.

To evaluate the performance of the logistic regression model, we calculated the cross-validation score using the Scikit-learn library. The cross-validation score measures the accuracy of the model in predicting the cluster membership of the data points, and the result obtained from the cross-validation scores are:

Cross-validation scores: [0.99427366 0.99367089 0.99367089 0.99306811 0.99427366]

Additionally, we calculated the homogeneity score to assess the extent to which each cluster contains data points that belong to the same class. we used Scikit-learn library to calculate the homogeneity score. the homogeneity score is:

Homogeneity score: 0.9462380425102818

Furthermore, we attempted to calculate the silhouette score to evaluate the quality of the clustering results. However, due to the large size of my dataset, we encountered a memory error when trying to calculate the silhouette score on the entire dataset. Therefore, we calculated the silhouette score on a subset of the data instead, and the calculated silhouette score is:

Silhouette score=0.2112602825625433

The same steps we applied for the K-means clustering algorithm, and we got the following results:

Cross-validation scores: [-8835.02649803, -6165.07854312, -6703.62842736, -6872.25926622, -3328.39587674]

Homogeneity score: 0.9085997012318614

Silhouette score= 0.455114130553408

Comparing the results from both algorithms we can say that:

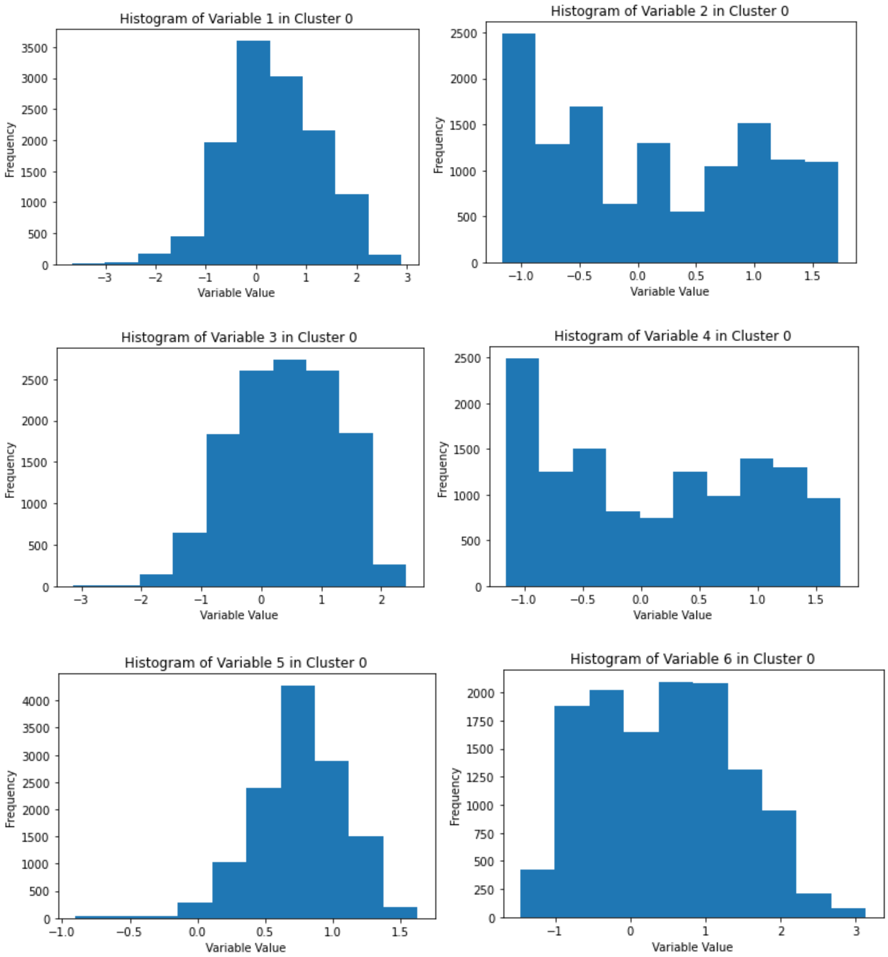
- DBSCAN was able to identify clusters with varying shapes and densities in my data, while K-Means was able to identify more spherical clusters.
- DBSCAN is generally better suited for datasets with irregular shapes and varying density, while K-Means is more suitable for datasets with well-defined spherical clusters.
- DBSCAN is able to identify the outliers, while K-means does not.
- The negative cross-validation scores of K-Means may indicate that the model is not performing well on my dataset.

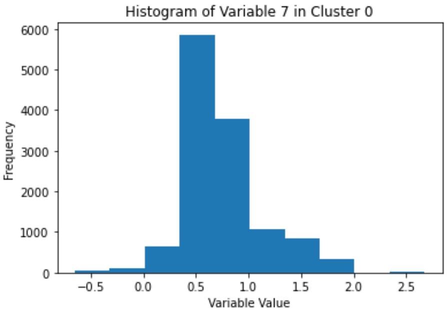
Based on the characteristics of my data, where the data contains outliers, we can say that the DBSCAN is more appropriate since it can handle noise and outliers better than k-means. So, in our case, DBSCAN is more appropriate choice as it can handle non-linear relationships between features and detect clusters with varying densities and shapes. It also does not require specifying the number of clusters beforehand, which can be useful when we don't have prior knowledge of the optimal number of clusters in my data.

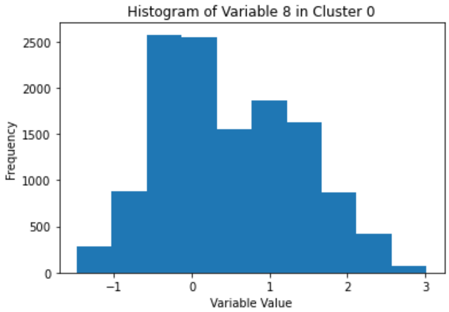
4.5. Perform Clustering on a Dataset and Visualize the Resulting Clusters

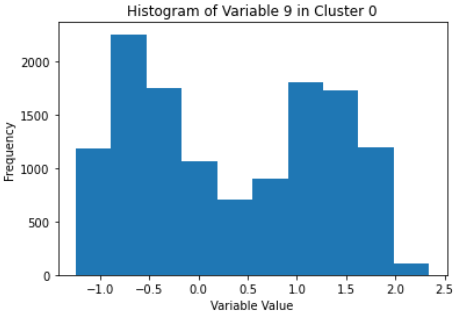
The histograms and boxplots were used to Perform clustering on a dataset and visualize the resulting clusters (using DBSCAN clustering algorithm), the histograms and boxplots created for each cluster provide a visual representation of the data within each cluster, allowing me to explore the distribution of each variable and identify any patterns or trends in the data. By calculating the maximum and minimum values of each variable within each cluster, we could gain additional insights into the characteristics of each cluster and identify any outliers or unusual data points.

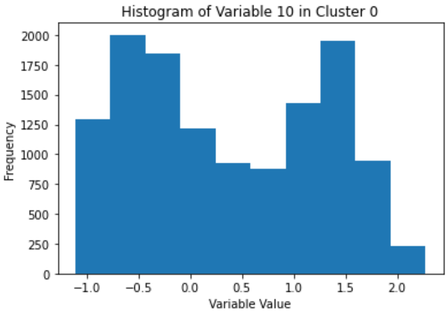
Cluster 0: *Figure 41: Cluster 0*

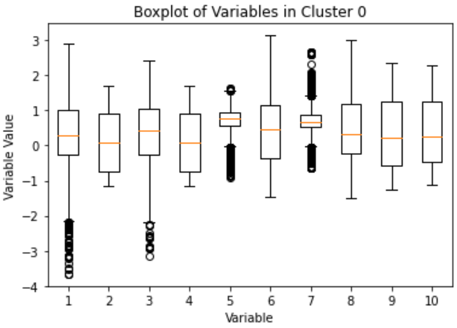












We can see from the previous figures of cluster (0), that the frequencies of each variable within the cluster number (0), which is representing the first big cluster, and we can see also that the majority of the outliers, that defined by DBSCAN clustering, are identified in this cluster and these outliers come mainly from the percentage of the fan valve sensors and the filter pressure sensors. This suggests that these variables are highly variable or have a high degree of variance within the dataset, relative to the other variables.

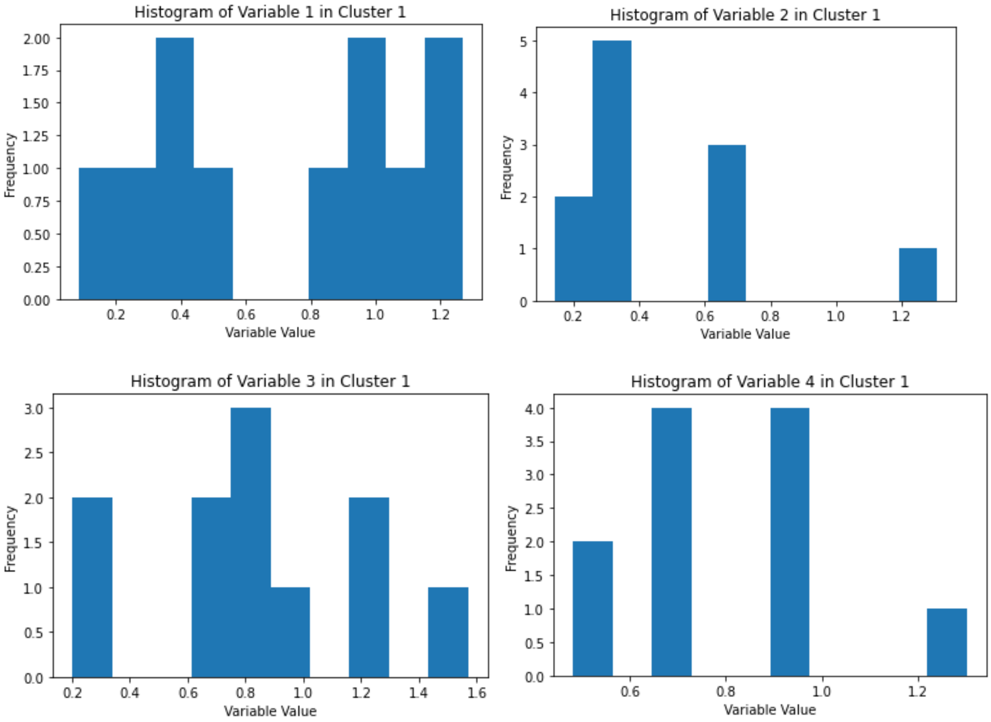
One possible interpretation of this finding is that the fan valve and filter pressure sensors may be particularly sensitive to changes or disruptions in the HVAC system. For example, if the fan valve is not functioning properly, this could lead to fluctuations in the measured values that are not representative of the true state of the system. Similarly, issues with the filters themselves (e.g., clogging or pressure imbalances) could affect the readings from the filter pressure sensors and lead to outliers in the data. So, the presence of outliers in this cluster suggests that these variables are important for predicting or identifying issues with the HVAC system.

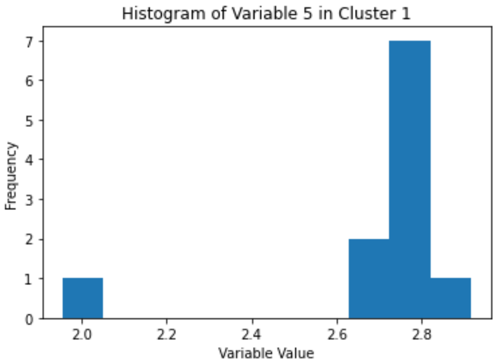
The maximum and minimum values of each variable in this cluster are shown in the table.

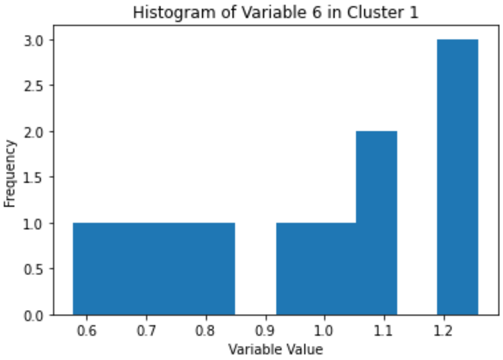
Table 11: The Maximum and Minimum Values of The Variable Cluster 0

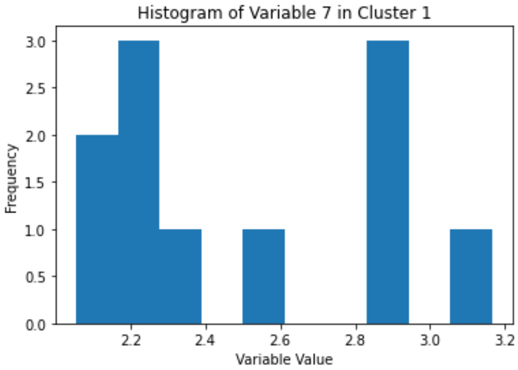
The variable	Max-value	Min-value
JV401_C	2.898837	-3.662319
JV401_D	1.717963	-1.158691
JV501_C	2.419102	-3.141263
JV501_D	1.712188	-1.156579
RD401_MV	1.629475	-0.904687
RD402_MV	3.137077	-1.463948
RD501_MV	2.672549	-0.653912
RD502_MV	3.018558	-1.481412
RF401_KV	2.343111	-1.246834
RF501_KV	2.274637	-1.111809

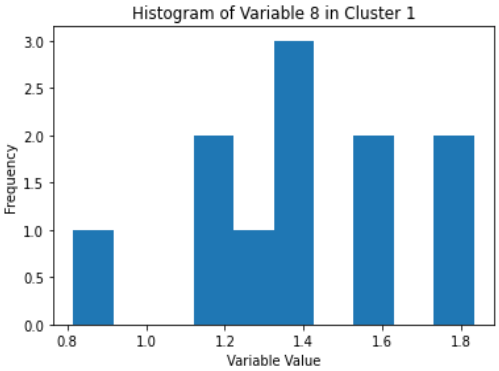
Cluster 1: *Figure 42: Cluster 1*

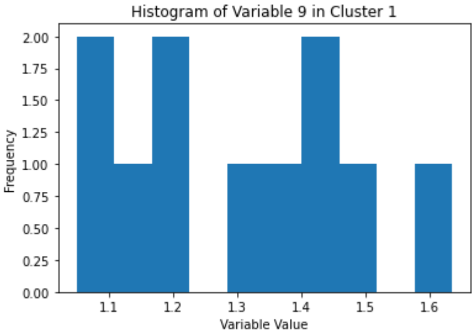


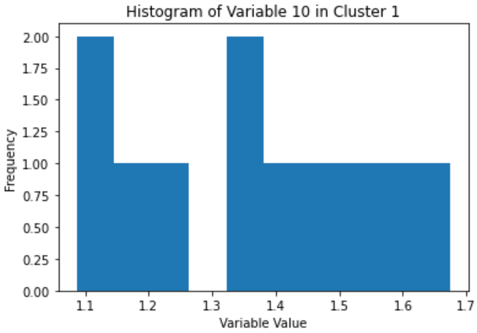


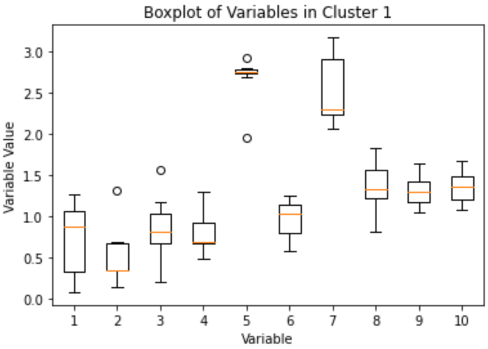












This small cluster represents a few numbers of observations that have a higher value than those which included in the previous cluster (cluster 0), and one interpretation of these sensor readings is, the HVAC system may have been running at a higher than usual airflow rate during those observations, causing a momentary increase in pressure drop across the filter. Alternatively, the pressure sensor itself could have been experiencing a temporary malfunction or interference that caused it to register an abnormal pressure reading.

And another interpretation is, the filter itself may have experienced a brief increase in pressure drop due to a sudden influx of dirt or particles, which then passed through the filter and the pressure drop returned to normal levels.

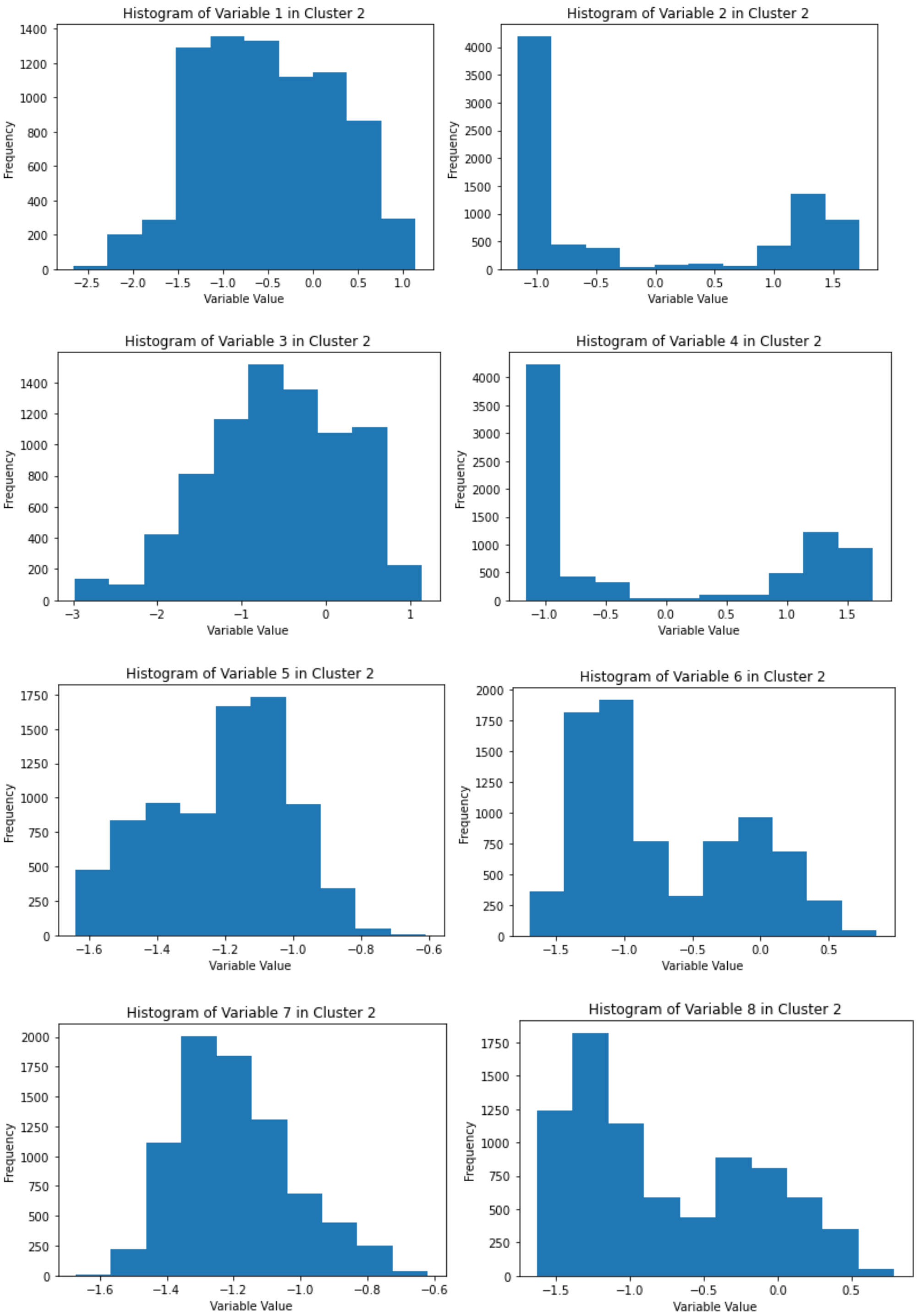
The maximum and minimum values of each variable in this cluster are shown in the table:

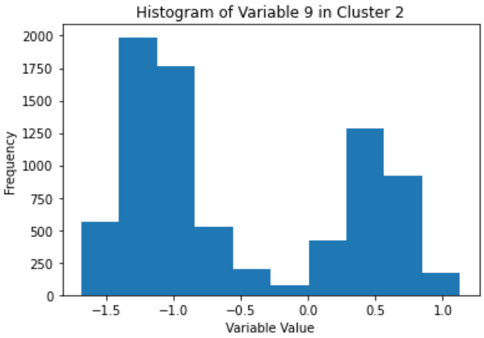
Table 12:The Maximum and Minimum Values of The Variable Cluster 1

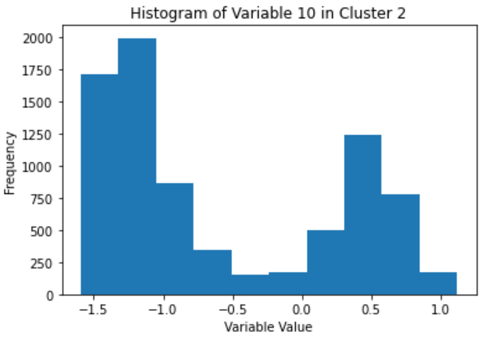
The variable	Max-value	Min-value
JV401_C	1.268704	0.084983
JV401_D	1.307012	0.142652
JV501_C	1.570051	0.201934
JV501_D	1.302364	0.482717
RD401_MV	2.916091	1.955494
RD402_MV	1.257932	0.578120
RD501_MV	3.165063	2.056722
RD502_MV	1.832436	0.815684
RF401_KV	1.635085	1.050466
RF501_KV	1.674760	1.087028

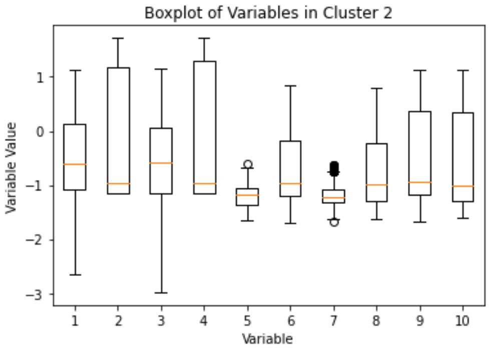
We can see that the min-value and the max-value for all variables in this cluster has positive values.

Cluster 2: *Figure 43: Cluster 2*









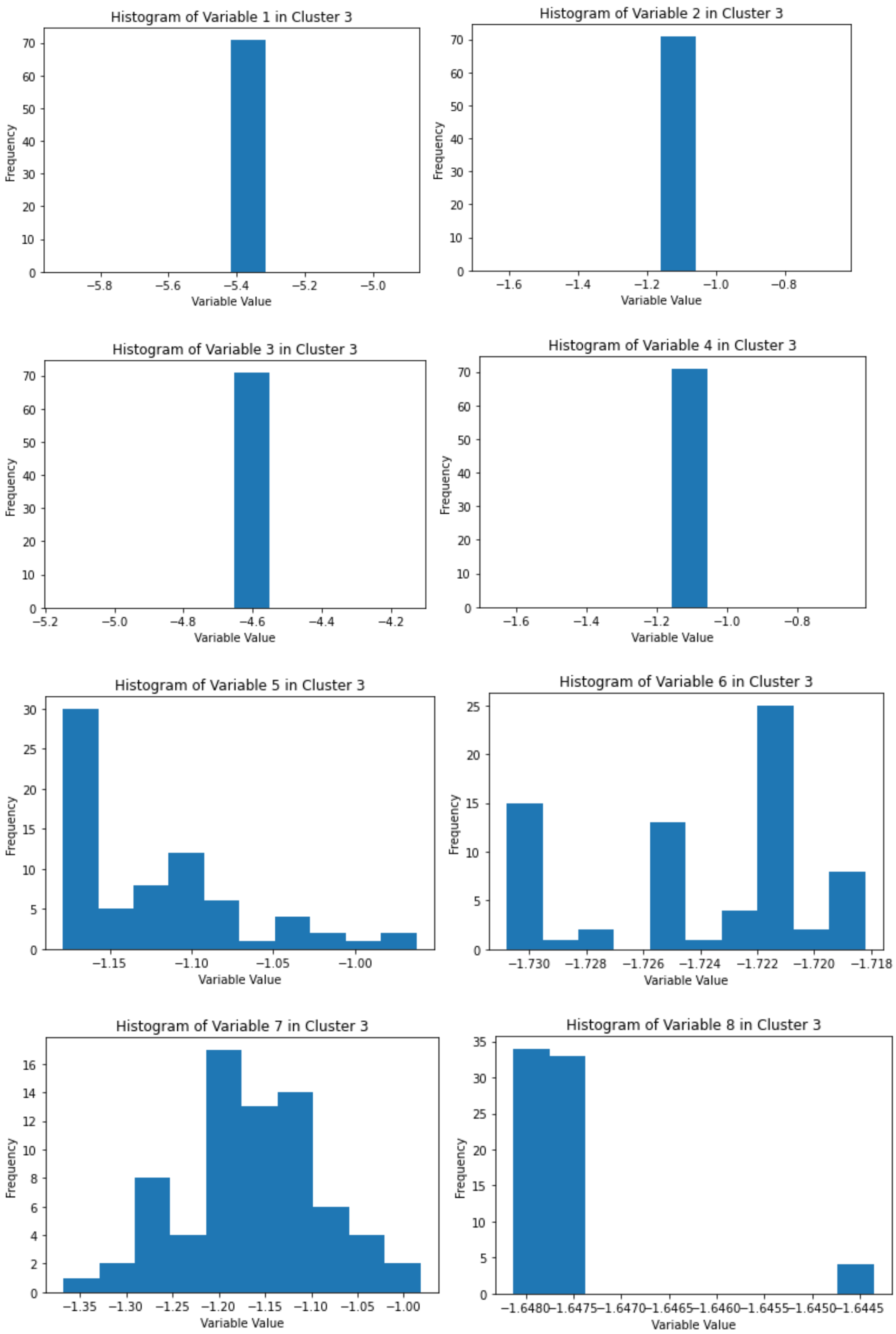
Comparing the main two clusters (0 and 2), it appears that they differ in terms of the ranges of the variables. For example, cluster (0) has a higher maximum value for most of the variables compared to cluster (2). And we can see a few points from the filter pressure sensors that defined as outliers.

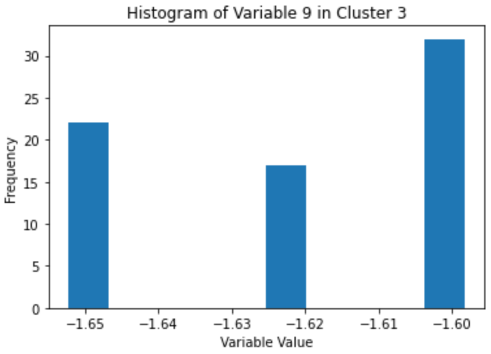
The maximum and minimum values of each variable in this cluster are shown in the table:

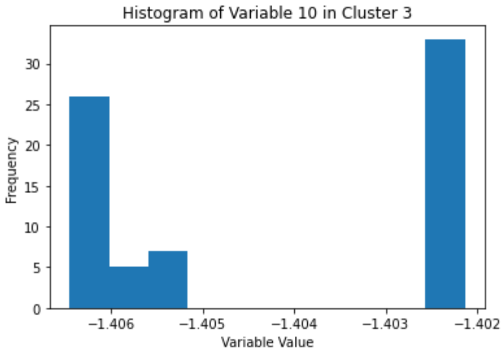
Table 13: The Maximum and Minimum Values of The Variable Cluster 2

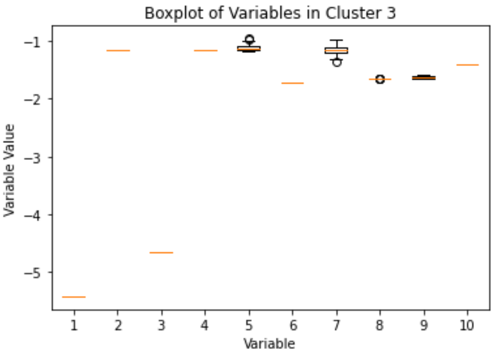
The variable	Max-value	Min-value
JV401_C	1.136175	-2.656901
JV401_D	1.717963	-1.158691
JV501_C	1.144442	-2.981351
JV501_D	1.712188	-1.156579
RD401_MV	-0.608150	-1.642129
RD402_MV	0.853274	-1.695624
RD501_MV	-0.620013	-1.671715
RD502_MV	0.789586	-1.629256
RF401_KV	1.128455	-1.685506
RF501_KV	1.119339	-1.594217

Cluster 3: *Figure 44: cluster 3*









We noticed that all the sensors in the HVAC system are producing small negative values that are smaller than the minimum values of the sensors, it is less likely to be a problem with individual sensors and more likely to be a systemic issue. And because we did not have enough information about the systems and all environmental factors at that specific period where the data has been collected, we could not know the real reasons for that, but we can give a few possibilities for why all sensors might be producing small negative values:

- HVAC system issues: There may be issues with the HVAC system itself that are causing negative values to be produced by all sensors. For example, the system may be running at too high or too low of a pressure, which could cause all sensors to produce negative values.
- Data transmission issues: If the negative values are being produced during data transmission, it is possible that there are issues with the data collection or transmission system that are causing the problem. For example, there could be a problem with the wiring or communication protocol that is causing the negative values to be produced.
- Calibration issues: It is possible that all sensors have calibration issues that are causing them to produce small negative values. This could be due to a variety

of reasons, such as the sensors not being properly calibrated during installation or due to drift over time.

- Environmental factors: Environmental factors, such as temperature or humidity, can also affect sensor readings. If the environment around the sensors is not stable or there are sudden changes in the environment, it could cause all sensors to produce negative values.
- Interference: Interference from other sources, such as electromagnetic interference (EMI) or radio frequency interference (RFI), can also cause all sensors to produce small negative values. This can happen when the sensors are installed in an environment with high levels of electrical or wireless activity. The interference can cause the sensors to pick up false signals, which can result in negative readings.
- Power supply issues: The sensors may be affected by power supply issues, such as low voltage or fluctuations in voltage. This can cause the sensors to produce inaccurate readings, including small negative values.
- Sensor failure: It is also possible that all sensors have failed and are producing small negative values as a result. This could happen if the sensors have reached the end of their lifespan or if they have been damaged due to improper handling or installation.

The maximum and minimum values of each variable in this cluster are shown in the table:

Table 14: The Maximum and Minimum Values of The Variable Cluster 3

The variable	Max-value	Min-value
JV401_C	-5.416285	-5.416285
JV401_D	-1.158691	-1.158691
JV501_C	-4.652571	-4.652571
JV501_D	-1.156579	-1.156579
RD401_MV	-0.962511	-1.178779
RD402_MV	-1.718225	-1.730777
RD501_MV	-0.982161	-1.367890
RD502_MV	-1.644356	-1.648130
RF401_KV	-1.598355	-1.652283
RF501_KV	-1.402135	-1.406452

We can see that the min-value and the max-value for all variables in this cluster has negative values. From both the histograms and the above values, some possible interpretations of the observations are:

1. Filter pressure: The negative values for the filter pressure in both the supply and return ducts suggest that there may be some blockage or restriction in the air flow, potentially due to a dirty or clogged filter.
2. Fan on/off status: The consistent negative values for the fan on/off status in both the supply and return ducts could indicate that the HVAC system is not functioning properly or that there is an issue with the sensors that detect the fan status.
3. Fan valve percentage: The negative values for the fan valve percentage in both the supply and return ducts suggest that the HVAC system may be trying to reduce the air flow, possibly due to a temperature or humidity control issue.
4. Fan pressure: The negative values for the fan pressure in both the supply and return ducts could also indicate that the HVAC system is not functioning properly or that there is an issue with the sensors that detect the fan pressure.
5. Air flow: The negative values for the air flow in both the supply and return ducts suggest that there may be a restriction in the air flow, potentially due to a dirty or clogged filter, as mentioned earlier.

Based on the analysis conducted, it is evident that predicting the time for filter replacement is feasible only if adequate data is available that represents the period where the filter is close to failing. This is due to the need for comprehensive information on the changes that occur in each component of the HVAC system as the filters become clogged. The critical sensors that aid in the prediction of filter failure time are the percentage of fan valve in both the supply and return ducts, filter pressure in both the supply and return ducts, and the airflow in both the supply and return ducts. These parameters have been identified as significant indicators in predicting the time of filter failure based on the previous analysis conducted.

4.6. The Hypothesizes:

To develop a machine learning model for predicting the time of filter failure, certain assumptions were made regarding the sensor readings that represent the state of clogged filters. The predictive algorithm used these readings as new data for training the model. The assumptions, were based on the assumption that all other factors that can potentially cause abnormal readings from the sensors were controlled for, including factors such as the surrounding environmental conditions, the effectiveness of the sensors, the transfer and storage of the data, and the operating modes of the system, including the potential impact of human intervention.

4.7. Supervised Machine learning model (The Predictive Model)

The supervised machine learning model implemented in this study involved using a decision tree regression algorithm to predict the time of filter failure in an HVAC system. Due to the absence of time of filter failure in the dataset, a proxy variable was defined as a substitute. The filter pressure was chosen as the proxy variable, as it is expected to increase as the filter clogs and eventually reaches a critical level that leads to filter failure. The decision tree regression model was trained to predict the filter pressure in the supply duct based on input features, which included pressure readings of fans in both supply and return ducts, filter pressure in the supply duct, on/off status of fans in both ducts, percentage of fan valve in both ducts, and air flow in both ducts.

To evaluate the performance of the model, the dataset was split into training and test sets, and a portion of the filter pressure data was included in the training set. Mean squared error was used as the performance metric for model evaluation.

The trained model was then used to predict the filter pressure in the supply duct for a new data point represented by new_data. The input features for this prediction were pressure readings of fans in both ducts, filter pressure in the supply duct, on/off status of fans in both ducts, percentage of fan valve in both ducts, and air flow in both ducts. The filter pressure in the return duct was used as the output variable.

By monitoring the predicted filter pressure over time, it is possible to schedule maintenance or replacement before filter failure occurs. When the predicted filter pressure reaches a critical level, this indicates that the filter is likely to fail soon, and necessary maintenance measures can be taken to prevent downtime.

4.8. Evaluate the Predictive Model

To provide a more comprehensive evaluation of the model, the Mean Squared Error (MSE), the Mean Absolute Error (MAE), the R-squared (R2) and Coefficient of Determination (r) are used to evaluate the performance of a predictive model.

1. Mean Squared Error (MSE): The code (A.24) used the trained machine learning model to predict the target variable for the test set X-test. The predicted values are stored in the y-pred variable. Then it calculated the Mean Squared Error (MSE) between the predicted values (y-pred) and the true values (y-test) of the target variable. The mean-squared-error () function is imported from the sklearn.metrics module, and as a result we got: MSE on the test set is 0.0499104481642639.

   This means that, on average, the squared difference between the predicted and true values of the target variable is 0.0499104481642639. where, a lower value of MSE indicates better performance of the model.

2. Mean Absolute Error (MAE): The code (A.25) calculates the MAE of the model by calling the mean_absolute_error function and passing it two arguments y_test and y_pred, where:
   - y_test is the true values of the target variable for the test set.
   - y_pred is the predicted values of the target variable for the test set.

The mean-absolute-error function calculates the mean absolute difference between the true values and the predicted values, and as result we got: Mean absolute error on test set: 0.12545839423472602.

That means the MAE on the test set is 0.12545839423472602, and this means that, on average, the model's predictions are off by about 0.127 units from the true values. A lower MAE indicates better performance, so this value suggests that the model is performing well.

3. R-squared (R2): The code (A.25) calculates the R-squared (R2) metric of a machine learning model on a test set using the scikit-learn library. Then the code calculates the R2 of the model by calling the r2_score function and passing it two arguments y_test and y_pred, where:

    - y_test is the true values of the target variable for the test set, and
    - y_pred is the predicted values of the target variable for the test set.

    The r2_score function calculates the R2 metric, which is a measure of how well the model fits the data. The R2 metric represents the proportion of the variance in the target variable that can be explained by the model, and as result I got:

    R-squared on test set: 0.9495453281555365. The R2 on the test set is 0.9495453281555365. This means that the model can explain about 95% of the variance in the target variable. A higher R2 indicates better performance, so this value also suggests that the model is performing well.

4. Coefficient of Determination (r): The code (A.25) calculated the predicted values of the target variable using the trained machine learning model. You can do this using the predict () function of the model, then we calculated the Pearson correlation coefficient between the predicted values (y_pred) and the true values (y_test) of the target variable by using the pearsonr () function from the scipy.stats module. the pearsonr() function returns two values: the calculated correlation coefficient and the associated p-value. In this case, we are only interested in the correlation coefficient, so we assigned it to the variable r and ignored the p-value using the underscore '_', and as a result, we got:
    Coefficient of Determination (r) on test set: 0.9748092963397764.
    A Coefficient of Determination (r) of 0.9748 indicates that the model's predictions are highly correlated with the true values of the target variable. This means that the model is doing a good job of capturing the underlying patterns in the data and making accurate predictions.
    The r value of 0.9748 suggests that the model is able to explain about 97.48% of the variation in the target variable, which is quite high. So, we can say that is

a very good result and indicates that the model is performing well on the test set.

So, the results we obtained suggest that the machine learning model is performing well on the test set. The model's predictions are on average quite close to the true values, and the model can explain a large proportion of the variance in the target variable.

4.9. An analysis of the limitations behind this study

In this study we faced some limitations:

1) Limitation of HVAC system type: The study was conducted on a regulator-type HVAC system, but the temperature variable was not available for analysis. This could limit the accuracy of the predictive maintenance model.
2) Limitation of insufficient data: The study was limited by the lack of sufficient data, especially during the period when filters are close to failing. This could limit the accuracy and reliability of the predictive maintenance model.
3) Limitation of data size: Due to the large size of the dataset, it was not possible to calculate the silhouette score for the entire dataset. As a result, the scores were only calculated for subsets of the data, which could affect the overall accuracy of the results.
4) Limitation of environmental factors: The study did not take into account the surrounding environmental factors that could affect the performance and longevity of the filters. This could limit the accuracy of the predictive maintenance model.
5) Limitation of maintenance schedule: The study did not have access to information about previous filter replacements or maintenance schedules, which could have provided valuable insights into the performance and lifespan of the filters.
6) Limitation of data collection time: The data used in the study was collected during the COVID-19 pandemic, which may have affected the performance and behavior of the HVAC system and may not reflect typical usage patterns. This could limit the generalizability of the study results.

5. Discussing the Results

5.1. The Questions This Thesis Tries to Answer

5.1.1. The First Question of This Study

As for the first question of this study (Can machine learning algorithms be effectively utilized for predictive maintenance in HVAC systems in building automation?), we note that:

in this study, we used several machine_learning algorithms, including PCA, DBSCAN, K-Means, and DecisionTreeRegressor, to predict the time of filter failure in HVAC systems. we evaluated the effectiveness of these algorithms by comparing their performance in predicting filter failure and assessing their accuracy and precision.

Our results showed that DBSCAN was particularly effective for my data, which did not have a regular shape and contained outliers. DBSCAN was able to identify clusters of similar data points and exclude outliers from these clusters, which helped improve the accuracy and precision of our predictions. In contrast, K-Means was less effective for our data, as it assumes a regular shape and struggles to handle outliers.

We also found that PCA was useful for reducing the dimensionality of our data and identifying the most important features for predicting filter failure. This allowed me to simplify my models and improve their interpretability, while still maintaining good predictive performance.

Finally, we used DecisionTreeRegressor to predict the time of filter failure based on the filter pressure as the output (the predicted variable) and other variables. Our results showed that this algorithm was able to accurately predict the time of filter failure, with a high level of precision.

Overall, our results demonstrate the effectiveness of machine learning algorithms for predictive maintenance in HVAC systems. We found that DBSCAN was particularly useful for the data, due to its ability to handle irregular shapes and outliers. However, we also acknowledge the limitations of these algorithms, such as the need for high-quality data and careful algorithm selection. Our findings suggest that machine learning algorithms have the potential to improve the efficiency and effectiveness of predictive maintenance in building automation and HVAC systems.

5.1.2. The Second Question of This Study

(Can predictive maintenance in building automation and HVAC systems improve energy efficiency and reduce the frequency of failures?):

The implementation of predictive maintenance in building automation and HVAC systems has the potential to significantly improve energy efficiency and reduce the frequency of failures. By using data analysis and machine learning algorithms to predict maintenance needs, potential issues can be identified and addressed before they cause major failures or inefficiencies in the system. This can lead to significant energy savings and improved system performance.

Predictive maintenance can help to identify patterns and trends in system performance that may indicate a need for maintenance or repairs. By addressing these issues proactively, energy consumption can be reduced, and the system can operate more efficiently. For example, by predicting the time of filter failure, a maintenance team can schedule replacement before the filter becomes clogged, reducing the pressure drop across the filter and improving airflow, which can lead to energy savings.

Furthermore, predictive maintenance can reduce the frequency of failures, which can lead to improved system reliability and reduced downtime. This can help to prevent costly emergency repairs and reduce overall maintenance costs. By identifying and addressing potential issues before they cause major failures, maintenance teams can prioritize resources and focus their efforts on the most critical needs.

In terms of cost savings, predictive maintenance can be more cost-effective than reactive maintenance, which involves repairing or replacing equipment after it has already failed. Reactive maintenance can lead to costly emergency repairs and downtime, while predictive maintenance can help to prevent these issues and reduce overall maintenance costs.

Overall, our findings suggest that the implementation of predictive maintenance in building automation and HVAC systems can lead to significant improvements in energy efficiency, reduced frequency of failures, and potential cost savings compared to reactive maintenance.

5.1.3 The Third Question of This Study

(What is the impact of predictive maintenance on the longevity and performance of filters in building automation HVAC systems?):

The impact of predictive maintenance on the longevity and performance of filters in HVAC systems is an important consideration in building automation. By predicting the time of filter failure, maintenance teams can replace filters before they become clogged, which can improve filter performance and increase filter lifespan.

Filters play a critical role in maintaining indoor air quality, and their performance can impact employees health and comfort. Clogged filters can reduce airflow, leading to increased energy consumption and reduced system performance. By replacing filters proactively, predictive maintenance can help to maintain optimal airflow and improve overall system performance.

Moreover, predictive maintenance can help to optimize filter lifespan by identifying the optimal time for replacement based on filter pressure and other factors. This can lead to longer filter lifespan and reduced waste, which can have important environmental implications. By reducing the frequency of filter replacements, predictive maintenance can also help to reduce maintenance costs and minimize downtime.

Finally, predictive maintenance can also have important implications for indoor air quality and employee's health. By maintaining optimal filter performance, predictive maintenance can help to reduce airborne pollutants and improve indoor air quality. Improved indoor air quality can have important implications for health and productivity and can lead to reduced absenteeism and improved overall well-being.

Overall, our findings suggest that the implementation of predictive maintenance in building automation and HVAC systems can have important implications for filter longevity and performance, indoor air quality, and employee's health and comfort. By proactively addressing maintenance needs, maintenance teams can optimize filter performance and improve overall system efficiency.

5.1.4 The Fourth Question of This Study

(How can the implementation of predictive maintenance improve the overall maintenance strategy in building automation and HVAC systems?):

The implementation of predictive maintenance can have a significant impact on the overall maintenance strategy in building automation and HVAC systems. By leveraging machine learning algorithms to predict equipment failures before they occur, predictive maintenance can help optimize maintenance schedules, reduce downtime, and improve system reliability.

One of the key advantages of predictive maintenance is the ability to identify potential issues before they cause major equipment failures. This enables maintenance teams to schedule maintenance activities proactively, reducing downtime and improving equipment uptime. Predictive maintenance can also help optimize maintenance schedules by identifying the optimal time for maintenance activities based on equipment usage and performance data. This can help reduce the overall cost of maintenance while ensuring equipment is operating at peak performance.

However, the implementation of predictive maintenance is not without challenges. One of the main challenges is data collection and processing. Predictive maintenance requires large volumes of data, which must be collected, processed, and analyzed in real-time. This requires sophisticated data collection and processing tools, as well as specialized expertise in data analysis and machine learning.

Another challenge is algorithm selection. There are many machine learning algorithms available for predictive maintenance, each with their own strengths and weaknesses. Selecting the right algorithm for a particular use case requires careful analysis and experimentation.

Finally, cost-benefit analysis is an important consideration when implementing predictive maintenance. While predictive maintenance can help reduce maintenance costs and improve equipment reliability, it also requires investment in data collection and processing tools, as well as specialized expertise in data analysis and machine learning. It is important to weigh the potential benefits against the costs when considering the implementation of predictive maintenance.

In summary, the implementation of predictive maintenance can have a significant impact on the overall maintenance strategy in building automation and HVAC systems. By enabling proactive maintenance scheduling and optimization, predictive maintenance can help reduce downtime, improve system reliability, and reduce the overall cost of maintenance. However, there are challenges to implementing predictive maintenance, including data collection and processing, algorithm selection, and cost-benefit analysis.

5.2. Evaluation Metrics

To interpret the cross-validation scores we consider the following:

the scores indicate the average performance of the model on different test sets that were created by splitting the data into different folds. In my case, the output of cross-validation scores shows that the model achieved high accuracy on all 5 folds, with scores ranging from 0.993 to 0.994. This indicates that the DBSCAN algorithm was able to accurately cluster the data.

The homogeneity score, on the other hand, indicates the degree to which each cluster contains only samples belonging to a single class. A score of 1.0 indicates perfectly homogeneous clusters, while a score of 0.0 indicates perfectly heterogeneous clusters. In my case, the homogeneity score of 0.946 suggests that the clusters produced by the DBSCAN algorithm contain mostly samples from a single class, indicating that the algorithm was successful in identifying clusters with similar properties.

In addition, we obtained a silhouette score of 0.288, which measures how similar each sample is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a higher score indicates that the samples are closer to their own clusters and farther from neighboring clusters. In our case, the silhouette score of 0.288 may be considered good because the big size of my dataset or maybe the clusters are inherently difficult to separate. That indicates that the clusters produced by the DBSCAN algorithm have overlapping boundaries and some samples may be assigned to incorrect clusters. While this score is not very high, it still suggests that the DBSCAN algorithm was able to find some structure in the data and group similar samples together.

6. Conclusions

In this study, we investigated the effectiveness of machine learning algorithms in predictive maintenance for HVAC systems in building automation. We noted how reprocessing the data is actually one of the key steps in the predictive maintenance and HVAC system analysis, as it helps to clean, transform, and prepare the data for further analysis, for this purpose we utilized PCA to reduce the dimensionality from 10 to 4. Then, we employed DBSCAN algorithm to analyze how the data within the various clusters evolve over time. To evaluate the DBSCAN results, we used supervised learning with logistic regression. Interestingly, DBSCAN performed well on the available data, as evidenced by the high cross-validation scores and the high homogeneity score.

However, the lower silhouette score indicates that there is some room for improvement.

To predict the filter failure, we analyzed the 'DecisionTreeRegressor' algorithm, using filter pressure as the predictive feature since the time of failure was unavailable.

Our results showed that machine learning algorithms can be effectively utilized for predictive maintenance in HVAC systems. Considering the first question posed at the beginning of the thesis, i.e., "Can machine learning algorithms be effectively utilized for predictive maintenance in HVAC systems in building automation?", the answer is thus positive.

By predicting filter failure in an HVAC system using a machine learning algorithm, our study provides strong evidence that machine learning can optimize maintenance schedules, reduce costs, and improve the overall performance of HVAC systems. Our findings also demonstrate that predictive maintenance in building automation and HVAC systems can improve energy efficiency and reduce the frequency of failures, and that answers the second thesis question "Can predictive maintenance in building automation and HVAC systems improve energy efficiency and reduce the frequency of failures?".

We also note by using data-driven algorithms to predict maintenance needs, building owners can optimize maintenance schedules, improve system performance, and prevent failures before they occur. These results are not only reduced energy consumption, as noted above, but also increased reliability of HVAC systems.

This study suggests thus that predictive maintenance can have a positive impact on the longevity and performance of filters in building automation HVAC systems. By predicting filter failure using a machine learning algorithm, building owners can replace them before they fail, resulting in improved air quality, reduced energy consumption, and longer filter lifespan, and this answers the third thesis question, "What is the impact of predictive maintenance on the longevity and performance of filters in building automation HVAC systems?".

The implementation of predictive maintenance can also improve the overall maintenance strategy in building automation and HVAC systems by optimizing maintenance schedules, reducing maintenance costs, and improving system performance. By using data-driven algorithms to predict maintenance needs, building owners can prioritize maintenance activities and prevent failures before they occur. This results in improved system reliability, reduced downtime, and increased energy efficiency, and this answers the last thesis question, "How can the implementation of predictive maintenance improve the overall maintenance strategy in building automation and HVAC systems?".

In conclusion, our study demonstrates that machine learning algorithms can be effectively utilized for predictive maintenance in HVAC systems in building automation. By using data-driven algorithms, building owners can optimize maintenance schedules, reduce costs, and improve the overall performance of HVAC systems. The implementation of predictive maintenance can also have a positive impact on the longevity and performance of filters in building automation HVAC systems and improve the overall maintenance strategy. Future research can explore additional machine learning techniques and applications to further enhance predictive maintenance in building automation and HVAC systems.

6.1. Future Research

There exist several issues that could be explored to further improve the capabilities of predictive maintenance of HVAC systems in building automation.

Firstly, one may try to evaluate the effectiveness of other machine learning algorithms, such as Random Forest and Support Vector Machines, and compare them to the algorithms analyzed in this study. Additionally, different clustering algorithms could be tested to determine their efficacy in identifying patterns and anomalies in the data.

Secondly, one may analyse the benefit of incorporating additional data sources into the predictive maintenance system. For example, weather data could be used to better estimate (and also predict) when the system has been or will be under heavier use.

Finally, the application of predictive maintenance techniques to other systems in building automation, such as lighting or security, could be explored too. Here one may try to determine the extent to which these techniques can be applied across different domains.

Overall, the results of this study suggest that predictive maintenance techniques using machine learning algorithms can be effective in improving the performance and efficiency of HVAC systems in building automation, and there is room for further research in this area.

In order to address the limitation that we faced in this study and thus further advance the field of predictive maintenance in building automation and HVAC systems, I would thus personally recommend the following future research topics:

1. Future research could focus on collecting more comprehensive data that includes temperature measurements. This could provide more accurate and detailed insights into the performance of the system and help to improve the accuracy of predictive maintenance models.
2. Future research could focus on collecting data specifically during the periods when the filters close to failure, or on developing techniques to extrapolate from existing data to predict filter failure more accurately.
3. Future research could focus on integrating additional data sources or developing more sophisticated predictive models that take into account a broader range of

factors to address the limitation of not having information on surrounding environmental factors or previous maintenance schedules.

4. Future research could focus on collecting data under more normal operating conditions to address the limitation of the data being collected during the COVID-19 pandemic to better understand how HVAC systems perform over a wider range of conditions.

These will very likely contribute to the development of more accurate and effective techniques for improving system performance and energy efficiency.

References

(MACC), M.-A. C. (2022, February 28). *What Is a Building Automation System?* Retrieved from MACC: https://info.midatlanticcontrols.com/blog/what-is-a-building-automation-system

Aamo, O. a. (2018, DECEMBER 12). 2018 Index IEEE Transactions on Automatic Control Vol. 63. *IEEE Transactions on Automatic Control*, 4453. Retrieved June 19, 2023, from http://ieeecss.org/sites/ieeecss/files/2019-10/Index_2018.pdf

Abdi, H. a. (2010). Wiley interdisciplinary reviews: computational statistics. *Wiley interdisciplinary reviews: computational statistics*, 433--459. Retrieved June 18, 2023, from https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101

Agency, U. E. (2023, APRIL 10). *Guide to Air Cleaners in the Home*. Retrieved from Agency, U.S. Environmental Protection: https://www.epa.gov/indoor-air-quality-iaq/guide-air-cleaners-home#tips

Agency, U. E. (2023, MARCH 13). *U.S. Environmental Protection Agency*. Retrieved from U.S. Environmental Protection Agency: https://www.epa.gov/indoor-air-quality-iaq/what-hepa-filter

Aiyi Liu, Y. Z. (2002, October 24). Block principal component analysis with application to gene microarray data classification. *Statistics in Medicine* . Retrieved June 2023, from https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1263

Altomonte, L. (2022, November 24). *Total Productive Maintenance*. Retrieved Jun 15, 2023, from Safty Culture: https://safetyculture.com/topics/total-productive-maintenance/

American Society of Heating, R. a.-C. (2017). *ASHRAE Handbook: HVAC Systems and Equipment.* Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE).

American Technical Publishers, I. P. (2009). *Building Automation System Integration with Open Protocols.* Retrieved from https://www.amazon.com/Building-Automation-Integration-Open-Protocols/dp/0826920128

Andreas C. Müller, S. G. (2017). *Introduction to Machine Learning with Python: A Guide for Data Scientists.* United State of America: "O'Reilly Media, Inc.". Retrieved from https://www.google.com/books?hl=no&lr=&id=1-4lDQAAQBAJ&oi=fnd&pg=PP1&dq=An+Introduction+to+Machine+Learning+with+Python%22+by+Andreas+M%C3%BCller+and+Sarah+Guido+(2016).+This+book+provides+a+good+introduction+to+machine+learning+in+Python,+including+a+cha

Ashtari Talkhestani, B. a. (2019). An architecture of an intelligent digital twin in a cyber-physical production system. *at-Automatisierungstechnik*, 762--782. Retrieved Jun 19, 2023, from https://www.degruyter.com/document/doi/10.1515/auto-2019-0039/html

avcontentteam. (2016, April 12). Tree Based Algorithms: A Complete Tutorial from Scratch (in R & Python). *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/

Beeck, V. C. (2021). A novel theory of Asian elephant high-frequency squeak production. *BMC biology*, 121. Retrieved from https://link.springer.com/article/10.1186/s12915-021-01026-z

Breiman, L. F. (1984). *Classification and regression trees.* Wadsworth Int.

Butler, R. (n.d.). Predictive Maintenance for HVAC Plays a Key Role in the Life of Your System. Retrieved from https://www.buildingsiot.com/blog/predictive-maintenance-for-hvac-plays-a-key-role-in-the-life-of-your-system-bd

Celina Gómez, C. J.-J.-S. (2019, September 1). Controlled Environment Food Production for Urban Agriculture. *ASHS Publications*, 1448–1458. Retrieved from https://journals.ashs.org/hortsci/view/journals/hortsci/54/9/article-p1448.xml

Chandola, V. (2009). Anomaly Detection for Discrete Sequences. *chandola2009anomaly*. Retrieved from https://www.academia.edu/download/44875573/Anomaly_Detection_for_Discrete_Sequences20160418-3991-1qkftu3.pdf

Chen, S. a. (2021). Integrating high share of renewable energy into power system using customer-sited energy storage. *Renewable and Sustainable Energy Reviews*, 110893. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364032121001878

DAVID J. KETCHEN, C. L. (1996, June). THE APPLICATION OF CLUSTER ANALYSIS IN STRATEGIC MANAGEMENT RESEARCH: AN ANALYSIS AND CRITIQUE. *Strategic Management Journal* . Retrieved Jun 19, 2023, from https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-0266(199606)17:6%3C441::AID-SMJ819%3E3.0.CO;2-G

Devanathan, K. a. (2021). An automated classification of HEp-2 cellular shapes using Bag-of-keypoint features and Ant Colony Optimization. *Biocybernetics and Biomedical Engineering*, 376--390. Retrieved Jun 17, 2023, from https://www.sciencedirect.com/science/article/abs/pii/S0208521621000140

developers, s.-l. (2007-2023). Clustering. *scikit-learn 1.2.2*.

developers, s.-l. (2007-2023). *sklearn.metrics.homogeneity_score*. Retrieved from scikit-learn 1.2.2: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html

Dzhambazov, G. B.-G. (2017). Predictive maintenance of building automation and HVAC systems. . *Energy and Buildings.*, 51-57.

Ellison, A. M. (1994). Right between the eyes--Visualizing Data by William S. Cleveland. *BioScience*, 622. Retrieved June 15, 2023, from https://search.proquest.com/openview/e112a2a1ac3cc2fbb0961675788da73c/1?pq-origsite=gscholar&cbl=34924

Emerson, J. W. (2013). The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 79--91. Retrieved June 17, 2023, from https://www.tandfonline.com/doi/abs/10.1080/10618600.2012.694762

Ester, M. a.-P. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (pp. 226--231).

Esteves, F. a. (2022). REVIEW OF ENERGY AUDIT AND BENCHMARKING TOOLS TO STUDY ENERGY EFFICIENCY THROUGH REDUCING CONSUMPTION IN WASTEWATER TREATMENT SYSTEMS. *Cadernos de Educa{\c{c}}{\~a}o Tecnologia e Sociedade*, 150--165. Retrieved June 19, 2023, from http://brajets.com/v3/index.php/brajets/article/view/996

G{\"o}nen, M. a. (2010). Localized multiple kernel regression. *IEEE*, 1425--1428. Retrieved Jun 2023, from https://ieeexplore.ieee.org/abstract/document/5597404/

Gbadamosi, A.-Q. a.-M. (2019, June 21). The role of internet of things in delivering smart construction. *CIB World Building Congress 2019*. Retrieved Jun 19, 2023, from https://uwe-repository.worktribe.com/preview/1492592/Gbadamosi%20et

GK. (2020, June 15). *GK inn på eiersiden i Piscada*. Retrieved from GK: https://www.gk.no/siste-nytt/2020/gk-inn-pa-eiersiden-i-piscada

Goodfellow, I. a. (2016). Deep feedforward networks. *Deep learning*. Retrieved Jun 16, 2023, from https://mnassar.github.io/deeplearninghandbook/slides/06_mlp.pdf

Goodfellow, I. a. (2016). *Deep learning.* MIT press. Retrieved Jun 2023, from https://books.google.com/books?hl=no&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=Goodfellow,+Bengio,+%26+Courville,+2016%3B+&ots=MNU4ivkHRU&sig=17TX8Q2n95JqcCdtjETSpmh-VtA

Han, J. a. (2022). *Data mining: concepts and techniques.* Morgan kaufmann. Retrieved Jun 17, 2023, from https://www.google.com/books?hl=no&lr=&id=NR1oEAAAQBAJ&oi=fnd&pg=PP1&dq=2.%09Han,+J.,+Kamber,+M.+and+Pei,+J.,+2011.+Data+mining:+concepts+and+techniques.+Elsevier&ots=_M9JLEpcr3&sig=IR-qelhrnTPQVWKkSkWiykYgx2w

Hannun, A. Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 65--69. Retrieved Jun 2023, from https://www.nature.com/articles/s41591-018-0268-3

Hastie, T. (2020). Ridge Regularization: An Essential Concept in Data Science. *Technometrics Vol. 62*, 426-433.

Hastie, T. a. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer.

Hastie, T. a. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer. Retrieved Jun 16, 2023, from https://link.springer.com/book/10.1007/978-0-387-21606-5

Hastie, T. a. (2009). *The elements of statistical learning: data mining, inference, and prediction.* New York, NY: Springer. Retrieved from https://link.springer.com/book/10.1007/978-0-387-21606-5

Ian Goodfellow, Y. B. (2016). *Deep Learning.* MIT press.

IBM. (2023, Jun 16). *What is predictive analytics?* Retrieved from IBM: https://www.ibm.com/topics/predictive-analytics#:~:text=Predictive%20analytics%20is%20a%20branch%20of%20advanced%20analytics,in%20this%20data%20to%20identify%20risks%20and%20opportunities.

International, B. (2023, February 6). BACnet International Publishes New BACnet Guide Specification. Retrieved from https://bacnetinternational.org/press-releases/bacnet-international-publishes-new-bacnet-guide-specification/

International, K. A. (2017, January). KNX, The worldwide STANDARD for home and building control. Retrieved from https://www.knx.org/wAssets/docs/downloads/Marketing/Presentations/HVAC-For-Manufacturers/HVAC-For-Manufacturers_en.pdf

Jain, A. K. (1999). Data clustering: a review. *ACM computing surveys (CSUR9*, 264--323. Retrieved from https://dl.acm.org/doi/abs/10.1145/331499.331504

James, G. W. (2013). An introduction to statistical learning. *Springer Link*. Retrieved from https://link.springer.com/book/10.1007/978-1-0716-1418-1

Ján Drgoňa, J. A. (2020). *All you need to know about model predictive control for buildings,.* Retrieved from https://www.sciencedirect.com/science/article/pii/S1367578820300584

Jayaprakash, S., Nagarajan, M., Prado, R., Subramanian, S., & Divakarachari, P. A. (2021). Review of Energy Management Strategies for Resource Allocation in the Cloud: Clustering, Optimization and Machine Learning. Retrieved from https://www.mdpi.com/1996-1073/14/17/5322

John D. Kelleher, B. T. (2018). *Data Science.* London, England: MIT Press. Retrieved from https://www.google.com/books?hl=no&lr=&id=UlpVDwAAQBAJ&oi=fnd&pg=PP7&dq=Applied+Machine+Learning%22+by+Kelleher+and+Tierney+(2018).+&ots=vVq_Tl788P&sig=r7vQmYM5YxRGt6V6vWYEBg6Ysw0

Jolliffe, I. T. (2002). *Principal component analysis for special types of data.* Springer.

Jolliffe, I. T. (2002). *Principal component analysis for special types of data.* New York, NY: Springer. Retrieved from https://link.springer.com/content/pdf/10.1007/0-387-22440-8_13.pdf

Jones, E. I. (2015). Total revenue and economic growth in Nigeria: Empirical evidence. *Emerging Trends in Educational Research and Policy Studies*, 40-46.

Joshi, S. S. (2023). Novel Correlation for Critical Speed for Solid Suspension in Stirred Tanks Developed Using Machine Learning Models Trained on Literature Data. *Industrial \&*

Engineering Chemistry Research. Retrieved from https://pubs.acs.org/doi/abs/10.1021/acs.iecr.3c00488

Jun Yuan, C. C. (2020, November 25). A survey of visual analytics techniques for machine learning. *Springer Nature*, 3–36. Retrieved Jun 17, 2023, from https://link.springer.com/article/10.1007/s41095-020-0191-7

Karunakaran, V. a. (2022). A non-invasive ultrasensitive diagnostic approach for COVID-19 infection using salivary label-free SERS fingerprinting and artificial intelligence. *Journal of Photochemistry and Photobiology B: Biology*, 112545. Retrieved from https://www.sciencedirect.com/science/article/pii/S1011134422001592

Ke, J. a. (2020). Data-driven predictive control of building energy consumption under the IoT architecture. *Wireless Communications and Mobile Computing*, 1--20. Retrieved June 18, 2023, from https://www.hindawi.com/journals/wcmc/2020/8849541/

Kelleher, J. D. (2018). *Data science.* MIT Press. Retrieved from https://books.google.com/books?hl=no&lr=&id=UlpVDwAAQBAJ&oi=fnd&pg=PP7&dq=Kelleher+%26+Tierney,+2018&ots=vVqZ_o6adK&sig=v5P_VbAEKoOYW8aB6Bv1-6h0MnE

Kelly, A. (2006). *Strategic Maintenance Planning.* Retrieved Jun 15, 2023, from https://books.google.no/books/about/Strategic_Maintenance_Planning.html?id=NuXalAEACAAJ&redir_esc=y

Kirk, A. (2012). *Data Visualization: a successful design process* (Packt publishing LTD ed.). Packt publishing LTD. Retrieved from https://books.google.no/books?hl=no&lr=&id=I4qBVLfD3t4C&oi=fnd&pg=PT6&dq=1.%09Kirk,+A.+(2016).+Data+visualization:+A+successful+design+process.+Packt+Publishing+Ltd.&ots=b7-NmKiE0r&sig=6zva73T-ksCjcWYYmjOtQKuEg0Y&redir_esc=y#v=onepage&q&f=false

Klein, P. a. (2019). Generation of Complex Data for AI-based Predictive Maintenance. *University of Trier*. Retrieved Jun 17, 2023, from https://www.researchgate.net/profile/Patrick-Klein-12/publication/335069998_Generation_of_Complex_Data_for_AI-based_Predictive_Maintenance_Research_with_a_Physical_Factory_Model/links/5d7a069fa6fdcc9961c1404a/Generation-of-Complex-Data-for-AI-based-Predic

Kohavi, R. (1995, August 20-25). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Kourou, K. (2015). Exarchos Th. P., Exarchos KP, Karamouzis MV, Fotiadis DI. *Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology J*, 8--17.

Kumar, A. (2022, April 16). Correlation Concepts, Matrix & Heatmap using Seaborn. *Data Analytics* . Retrieved June 16, 2023, from https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/

Kuo, Y.-H. a. (2019). From data to big data in production research: the past and future trends. *International Journal of Production Research*, 4828--4853. Retrieved June 18, 2023, from https://www.tandfonline.com/doi/abs/10.1080/00207543.2018.1443230

Laskin, M. a. (2021). Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*. Retrieved Jun 17, 2023, from https://arxiv.org/abs/2110.15191

Li, T. Z. (2021, October 20). Probabilistic graphical models in energy systems: A review. *SPRINGER LINK*, 699–728. Retrieved Jun 17, 2023, from https://link.springer.com/article/10.1007/s12273-021-0849-9

Libbrecht, M. W. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 321--332. Retrieved Jun 2023, from https://www.nature.com/articles/nrg3920

Linoff, G. S. (2011). *Data mining techniques: for marketing, sales, and customer relationship management.* John Wiley & Sons.

LLC, F. K. (2023). *Fiberglass Air Filters vs. Pleated Air Filters*. Retrieved from Filter King LLC: https://filterking.com/hvac-filters/fiberglass-air-filters-vs-pleated

LLumin. (2023). Predictive Maintenance Cost Savings: Reduce Your Maintenance Costs. Retrieved from https://llumin.com/predictive-maintenance-cost-savings-reduce-your-maintenance-costs-llu/

Lower, A. (2023). *What Is An Electrostatic Air Filter?* Retrieved from Second nature: https://www.secondnature.com/blog/electrostatic-air-filters

Luke, T. (2022, April 11). Create a K-Means Clustering Algorithm from Scratch in Python. *Medium*.

Mahsa Shoaran, B. A. (2018, June 07). Energy-Efficient Classification for Resource-Constrained Biomedical Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*. Retrieved Jun 16, 2023, from https://ieeexplore.ieee.org/abstract/document/8374841/authors#authors

Mangla, M. a. (2022). *Handbook of Research on Machine Learning: Foundations and Applications.* CRC Press. Retrieved from https://www.google.com/books?hl=no&lr=&id=2flwEAAAQBAJ&oi=fnd&pg=PT10&dq=PCA+(Principal+Component+Analysis)+in+Machine+Learning%22+by+Himanshu+Sharma,+Analytics+Vidhya,+2020,&ots=V7TBgwCkPI&sig=52tGZ6rdeDd4z89A3TrsrDLnulo

Martens, H. a. (2001). *Multivariate analysis of quality: an introduction.* John Wiley \& Sons.

Mastrandrea, G. (2022, Jul 6). *Correlation Matrix, Demystified*. Retrieved from Towards Data Science: https://towardsdatascience.com/correlation-matrix-demystified-3ae3405c86c1

MathWorks. (2021). *Statistics and Machine Learning Toolbox*. Retrieved from MathWorks: https://au.mathworks.com/help/stats/scatterplot-matrix.html

McKinney, W. a. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (pp. 51--56). Austin, TX. Retrieved from https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf

Meilundefined, M. (2005). Comparing Clusterings: An Axiomatic View. *Digital Library*. Retrieved June 15, 2023, from https://dl.acm.org/doi/abs/10.1145/1102351.1102424

Mendenhall, W. B. (2012). *Introduction to probability and statistics.* Cengage Learning. Retrieved from https://books.google.com/books?hl=no&lr=&id=fQsKAAAAQBAJ&oi=fnd&pg=PR3&dq=2.%09Mendenhall,+W.,+Beaver,+R.+J.,+%26+Beaver,+B.+M.+(2013).+Introduction+to+probability+and+statistics.+Cengage+Learning.&ots=5ZkqeN82_S&sig=mjwjOjUJLJLkTlV8Rr3qciY-dUM

MengYang Liu, M. L. (2022, Jun 6). The Application of the Unsupervised Migration Method Based on Deep Learning Model in the Marketing Oriented Allocation of High Level Accounting Talents. *National Center for Biotechnology Information*. Retrieved Jun 17, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9192229/

Mishra, S. (2017, May 19). Unsupervised Learning and Data Clustering. *Towards Data Science*. Retrieved Jun 17, 2023, from https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a

Mobley, K. (2013). *Maintenance Engineering Handbook.* Retrieved Jun 15, 2023, from https://books.google.no/books/about/Maintenance_Engineering_Handbook_Eighth.html?id=usPbAgAAQBAJ&redir_esc=y

Neeraj, N. K. (2020, July 8). A REVIEW ON MACHINE LEARNING (FEATURE SELECTION, CLASSIFICATION AND CLUSTERING) APPROACHES OF BIG DATA MINING IN DIFFERENT AREA OF RESEARCH. *JOURNAL OF CRITICAL REVIEWS*, vol.7.

Nielsen, A. (2019). *Practical time series analysis: Prediction with statistics and machine learning.* O'Reilly Media. Retrieved from https://books.google.com/books?hl=no&lr=&id=odCwDwAAQBAJ&oi=fnd&pg=PR2&dq=Aileen+Nielsen,+2016,+Data+Science+Central&ots=OVhmwfgguT&sig=uynvaTaSrSRL0uEq0NDPdTRvo1U

Nita Yodo, T. A. (2022, October 18). Condition-based monitoring as a robust strategy towards sustainable and resilient multi-energy infrastructure systems. *Sustainable and Resilient Infrastructure*, 170-189 . Retrieved Jun 16, 2023, from https://www.tandfonline.com/doi/citedby/10.1080/23789689.2022.2134648?scroll=top&needAccess=true&role=tab

Ozgode Yigin, B. a. (2023). Effect of distance measures on confidences of t-SNE embeddings and its implications on clustering for scRNA-seq data. *Scientific Reports*, 6567. Retrieved June 19, 2023, from https://www.nature.com/articles/s41598-023-32966-x

Patil, I. (2021). Visualizations with statistical details: The'ggstatsplot'approach. *Journal of Open Source Software*, 3167. Retrieved June 19, 2023, from https://joss.theoj.org/papers/10.21105/joss.03167.pdf

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 559--572. Retrieved June 19, 2023, from https://www.tandfonline.com/doi/pdf/10.1080/14786440109462720

Richie-Halford A, N. M. (2022, Jul 07). Groupyr: Sparse Group Lasso in Python. *PubMed Centra,* .

Rousseeuw, P. J. (1987, November). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, Pages 53-65. Retrieved Jun 16, 2023, from https://www.sciencedirect.com/science/article/pii/0377042787901257

S.Raji, R. (1998). End-to-End Solutions With LonWorks Control Technology . Retrieved from http://downloads.echelon.com/support/documentation/papers/end2end.pdf

Sacc{\`a}, V. a. (2019). Evaluation of machine learning algorithms performance for the prediction of early multiple sclerosis from resting-state FMRI connectivity data. *Brain imaging and behavior*, 1103--1114. Retrieved June 16, 2023, from https://link.springer.com/article/10.1007/s11682-018-9926-9

Sander, J. a.-P. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 169--194. Retrieved from https://link.springer.com/article/10.1023/A:1009745219419

Selamat, H. (2020). Review on HVAC System Optimization Towards. *International Energy Journal*. Retrieved from https://www.thaiscience.info/Journals/Article/IENJ/10992379.pdf

Shalev-Shwartz, S. a.-D. (2014). *Understanding machine learning: From theory to algorithms.* Cambridge university press.

Sharma, P. (2019, August 19). The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications. *Analytics Vidhya App for the Latest blog/Article*.

Shukor, N. A. (2015). An examination of online learning effectiveness using data mining. *Procedia-Social and Behavioral Sciences*, 555--562. Retrieved Jun 17, 2023, from https://www.sciencedirect.com/science/article/pii/S1877042815004395

Sutton, R. S. (2018). *Reinforcement learning: An introduction.* MIT press.

Tan, P.-N. a. (2006). Data. *Introduction to data mining*, 58--59. Retrieved from http://snap.stanford.edu/class/cs224w-2011/proj/cktan_Finalwriteup_v1.pdf

Tan, P.-N. a. (2006). Data. *Introduction to data mining*, 58--59. Retrieved Jun 13, 2023, from http://snap.stanford.edu/class/cs224w-2011/proj/cktan_Finalwriteup_v1.pdf

Tauzin, G. a.-M. (2021). Giotto-Tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration. *J. Mach. Learn. Res.*, 1532-4435.

Telea, A. C. (2014). *Data visualization: principles and practice.* London, New York: CRC Press. Retrieved June 15, 2023, from https://www.google.com/books?hl=no&lr=&id=AGjOBQAAQBAJ&oi=fnd&pg=PP1&dq=3.%09Telea,+A.+C.+(2014).+Data+visualization:+Principles+and+practice.+CRC+Press.&ots=NlCvvhZUIs&sig=obM2XOz1G2EC4lzJvmhgmwp9-4A

Trevor Hastie, J. F. (2001). *The Elements of Statistical Learning.* Springer Link.

Urvashi. (2018, August 23). *3 things to consider when implementing a building automation system (BAS)*. Retrieved from RCR Wierless News: https://www.rcrwireless.com/20180823/network-infrastructure/inbuildingtech/bas-building-automation-system

VinayakGoyal. (2021, April 13). *What is Unsupervised learning*. Retrieved Jun 17, 2023, from Data Science Central: https://www.datasciencecentral.com/what-is-unsupervised-learning-1/

Wenlong Fu, K. W. (2018, December 14). Vibration trend measurement for a hydropower generator based on optimal variational mode decomposition and an LSSVM improved with chaotic sine cosine algorithm optimization. *Measurement Science and Technology*. Retrieved Jun 14, 2023, from https://iopscience.iop.org/article/10.1088/1361-6501/aaf377/meta

Wickham, H. a. (2016). Data analysis. *ggplot2: elegant graphics for data analysis*, 189--20. Retrieved June 15, 2023, from https://link.springer.com/chapter/10.1007/978-3-319-24277-4_9

Wilke, C. O. (2019). *Fundamentals of data visualization: a primer on making informative and compelling figures.* O'Reilly Media. Retrieved from https://www.google.com/books?hl=no&lr=&id=XmmNDwAAQBAJ&oi=fnd&pg=PP1&dq=-%09Wilke,+C.+O.+(2019).+Fundamentals+of+Data+Visualization:+A+Primer+on+Making+Informative+and+Compelling+Figures.+O%27Reilly+Media,+Inc.&ots=6LO6jY2Cs7&sig=FlEUSosVBTtYHXStG6rmZboDU

Witten, I. H. (2002). Data mining: practical machine learning tools and techniques. *Acm Sigmod Record*, 76--77. Retrieved Jun 14, 2023, from https://dl.acm.org/doi/pdf/10.1145/507338.507355

Witten, I. H. (2002). Data mining: practical machine learning tools and techniques . *Acm Sigmod Record*, 76--77. Retrieved Jun 14, 2023, from https://dl.acm.org/doi/pdf/10.1145/507338.507355

Witten, I. H. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *Acm Sigmod Record*, 76--77. Retrieved Jun 16, 2023, from https://dl.acm.org/doi/pdf/10.1145/507338.507355

Wold, S. a. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 37--52. Retrieved June 18, 2023, from https://www.sciencedirect.com/science/article/pii/0169743987800849

Xianfei Yin, H. L.-H. (2019, May). Building information modelling for off-site construction: Review and future directions. *Automation in Construction*, 72-91. Retrieved Jun 15, 2023, from https://www.sciencedirect.com/science/article/pii/S0926580518310410

Xianfei Yin, H. L.-H. (2019, May). Building information modelling for off-site construction: Review and future directions. *Automation in Construction*, 72-91. Retrieved Jun 17, 2023, from https://doi.org/10.1016/j.autcon.2019.01.010.

Ye Yao, Y. Y. (2017). *Modeling and Control in Air-conditioning Systems.* Retrieved from https://link.springer.com/book/10.1007/978-3-662-53313-0

Yongyi Ran, X. Z. (2019, December 12). *A Survey of Predictive Maintenance: Systems, Purposes and Approaches*. Retrieved Jun 16, 2023, from Cornell University: https://arxiv.org/abs/1912.07383

Zhang, W. a. (2020). DeepHealth: A self-attention based method for instant intelligent predictive maintenance in industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 5461--5473. Retrieved June 18, 2023, from https://ieeexplore.ieee.org/abstract/document/9216077/

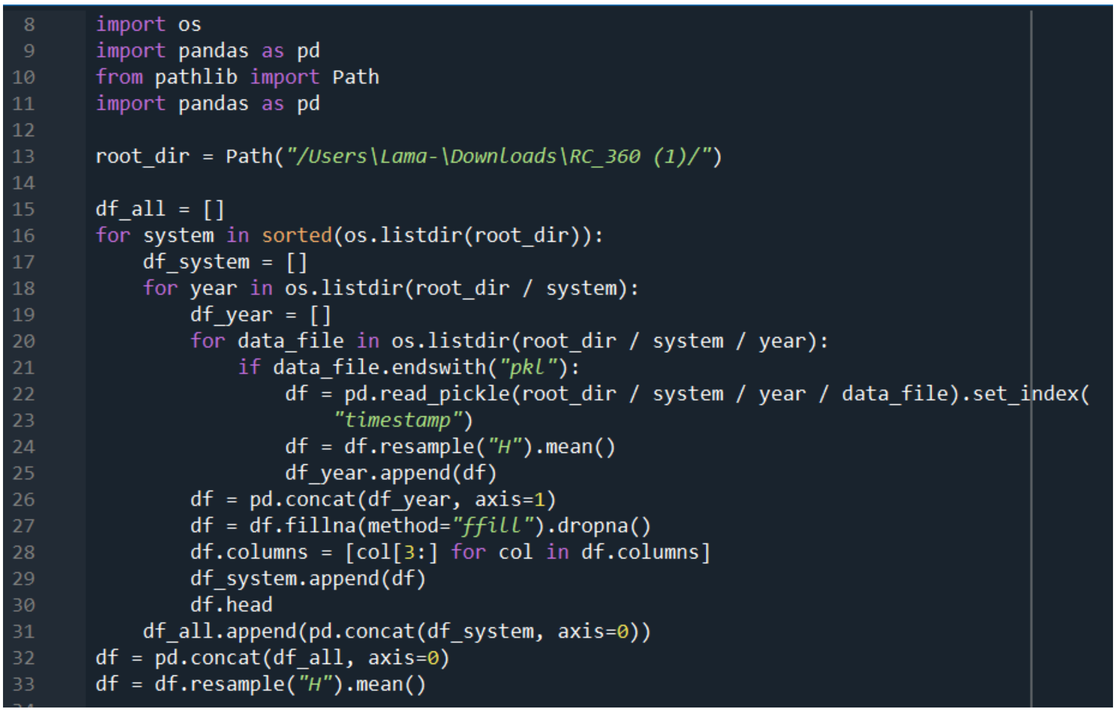
Zhang, Y.-P. a.-H.-T.-H. (2019). Psychometric testing of the evidence-based practice nursing leadership scale and the work environment scale after cross-cultural adaptation in mainland China. *Evaluation \& the Health Professions*, 328--343. Retrieved Jun 17, 2023, from https://journals.sagepub.com/doi/pdf/10.1177/0163278718801439

Appendix

Here are all the codes I used in python, which helped me reach the results of this study.

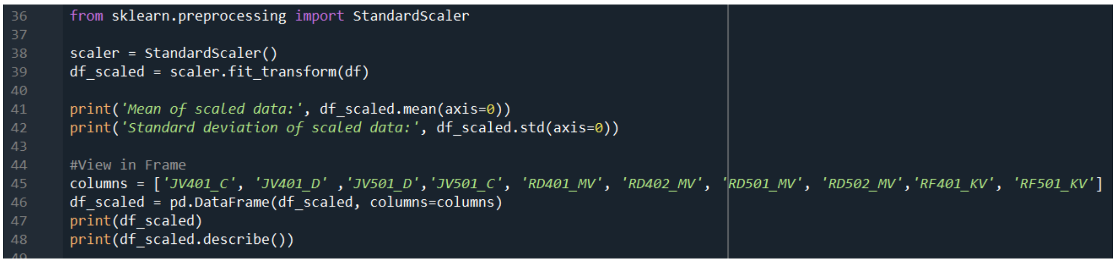
A Codes

A.1. Import and Combine the Data Files



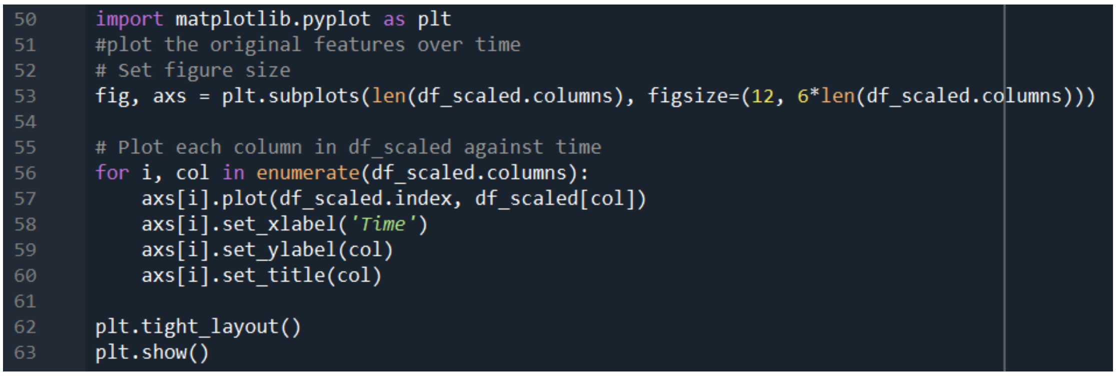
```
import os
import pandas as pd
from pathlib import Path
import pandas as pd

root_dir = Path("/Users\Lama-\Downloads\RC_360 (1)/")

df_all = []
for system in sorted(os.listdir(root_dir)):
    df_system = []
    for year in os.listdir(root_dir / system):
        df_year = []
        for data_file in os.listdir(root_dir / system / year):
            if data_file.endswith("pkl"):
                df = pd.read_pickle(root_dir / system / year / data_file).set_index(
                    "timestamp")
                df = df.resample("H").mean()
                df_year.append(df)
        df = pd.concat(df_year, axis=1)
        df = df.fillna(method="ffill").dropna()
        df.columns = [col[3:] for col in df.columns]
        df_system.append(df)
        df.head
    df_all.append(pd.concat(df_system, axis=0))
df = pd.concat(df_all, axis=0)
df = df.resample("H").mean()
```

A.2. Standardize the Data



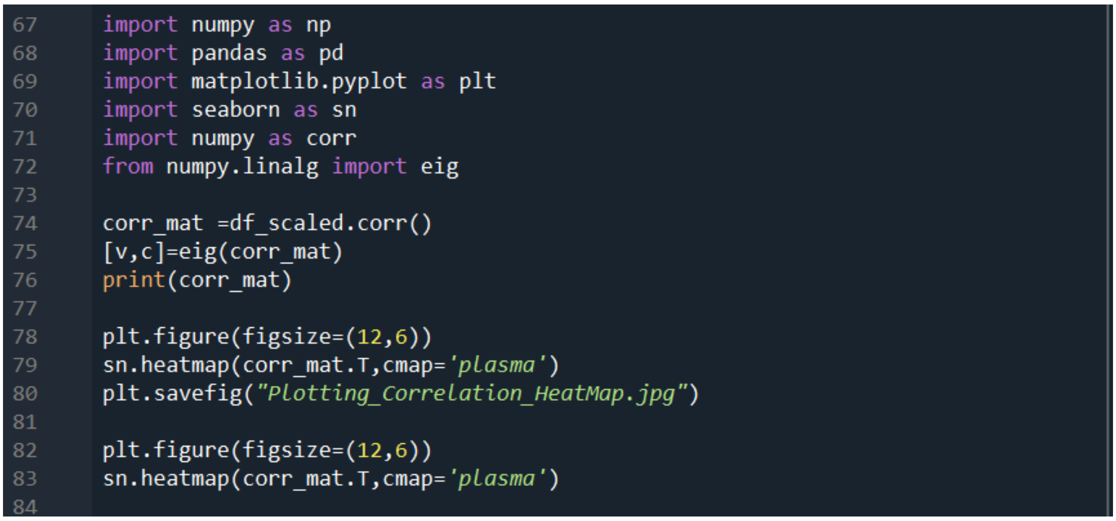
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

print('Mean of scaled data:', df_scaled.mean(axis=0))
print('Standard deviation of scaled data:', df_scaled.std(axis=0))

#View in Frame
columns = ['JV401_C', 'JV401_D' ,'JV501_D','JV501_C', 'RD401_MV', 'RD402_MV', 'RD501_MV', 'RD502_MV','RF401_KV', 'RF501_KV']
df_scaled = pd.DataFrame(df_scaled, columns=columns)
print(df_scaled)
print(df_scaled.describe())
```

A.3 Plot the Features with Time



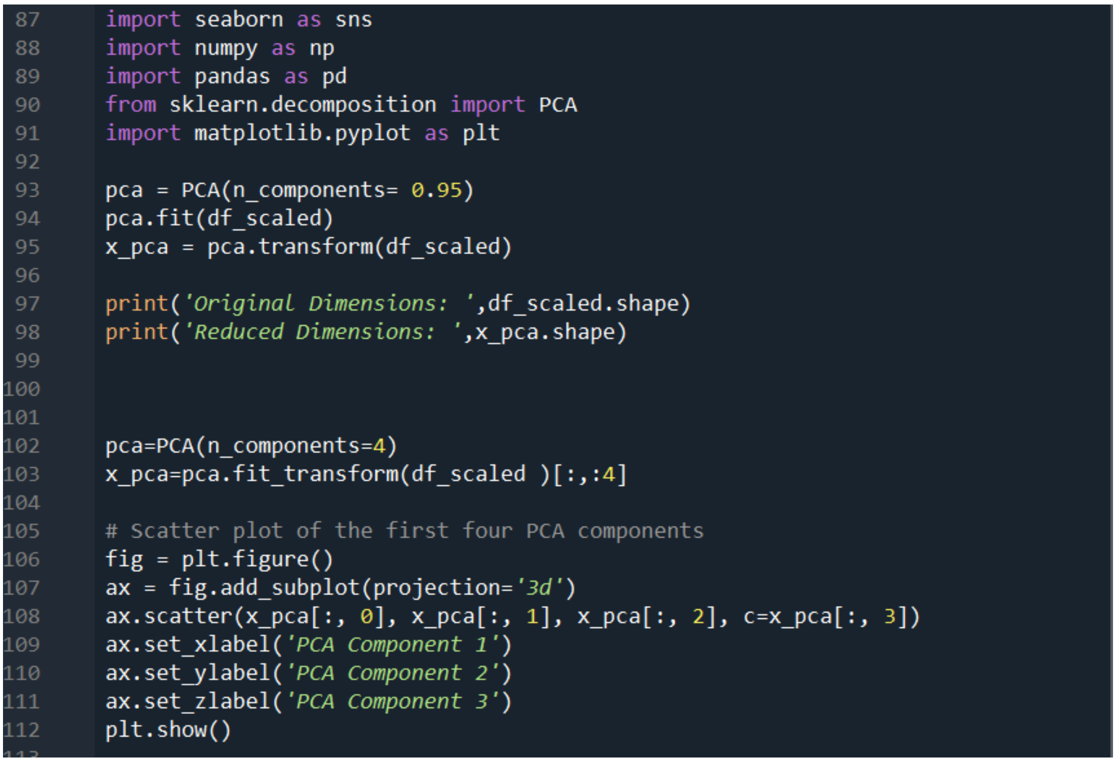
```python
import matplotlib.pyplot as plt
#plot the original features over time
# Set figure size
fig, axs = plt.subplots(len(df_scaled.columns), figsize=(12, 6*len(df_scaled.columns)))

# Plot each column in df_scaled against time
for i, col in enumerate(df_scaled.columns):
    axs[i].plot(df_scaled.index, df_scaled[col])
    axs[i].set_xlabel('Time')
    axs[i].set_ylabel(col)
    axs[i].set_title(col)

plt.tight_layout()
plt.show()
```

A.4 Correlation Matrix and Heatmap



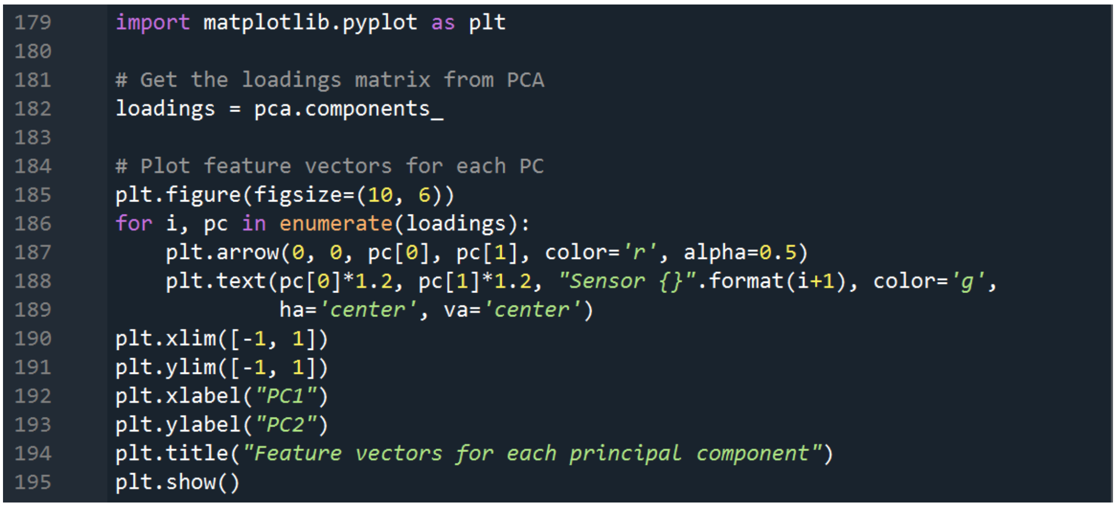
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import numpy as corr
from numpy.linalg import eig

corr_mat =df_scaled.corr()
[v,c]=eig(corr_mat)
print(corr_mat)

plt.figure(figsize=(12,6))
sn.heatmap(corr_mat.T,cmap='plasma')
plt.savefig("Plotting_Correlation_HeatMap.jpg")

plt.figure(figsize=(12,6))
sn.heatmap(corr_mat.T,cmap='plasma')
```

A.5 Implementation of PCA



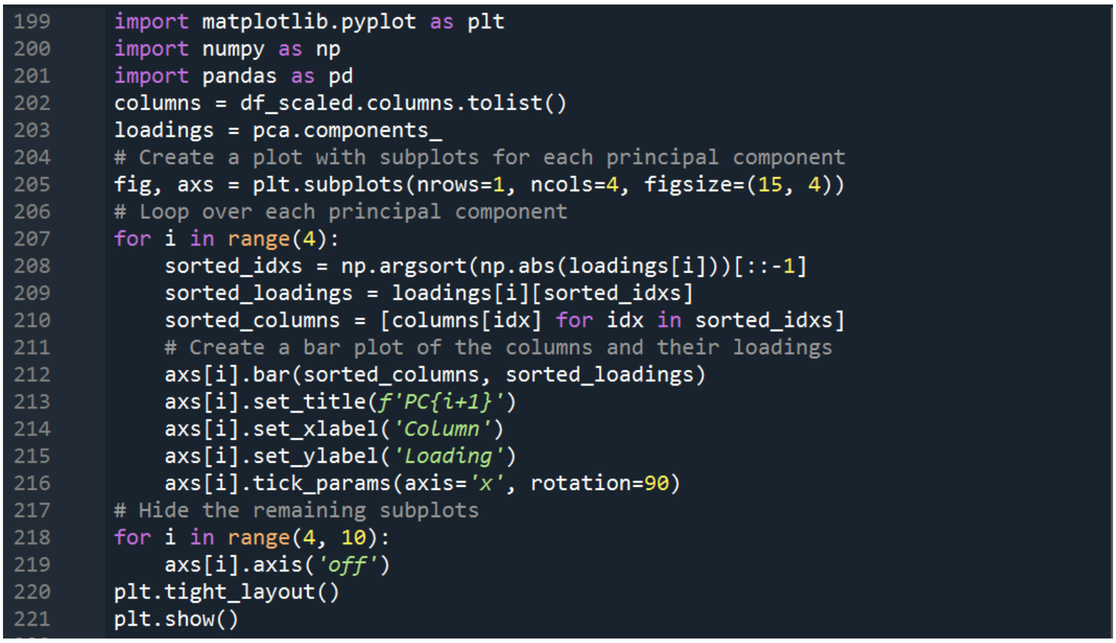
```python
import seaborn as sns
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

pca = PCA(n_components= 0.95)
pca.fit(df_scaled)
x_pca = pca.transform(df_scaled)

print('Original Dimensions: ',df_scaled.shape)
print('Reduced Dimensions: ',x_pca.shape)



pca=PCA(n_components=4)
x_pca=pca.fit_transform(df_scaled )[:,:4]

# Scatter plot of the first four PCA components
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(x_pca[:, 0], x_pca[:, 1], x_pca[:, 2], c=x_pca[:, 3])
ax.set_xlabel('PCA Component 1')
ax.set_ylabel('PCA Component 2')
ax.set_zlabel('PCA Component 3')
plt.show()
```

A.6 The Loadings



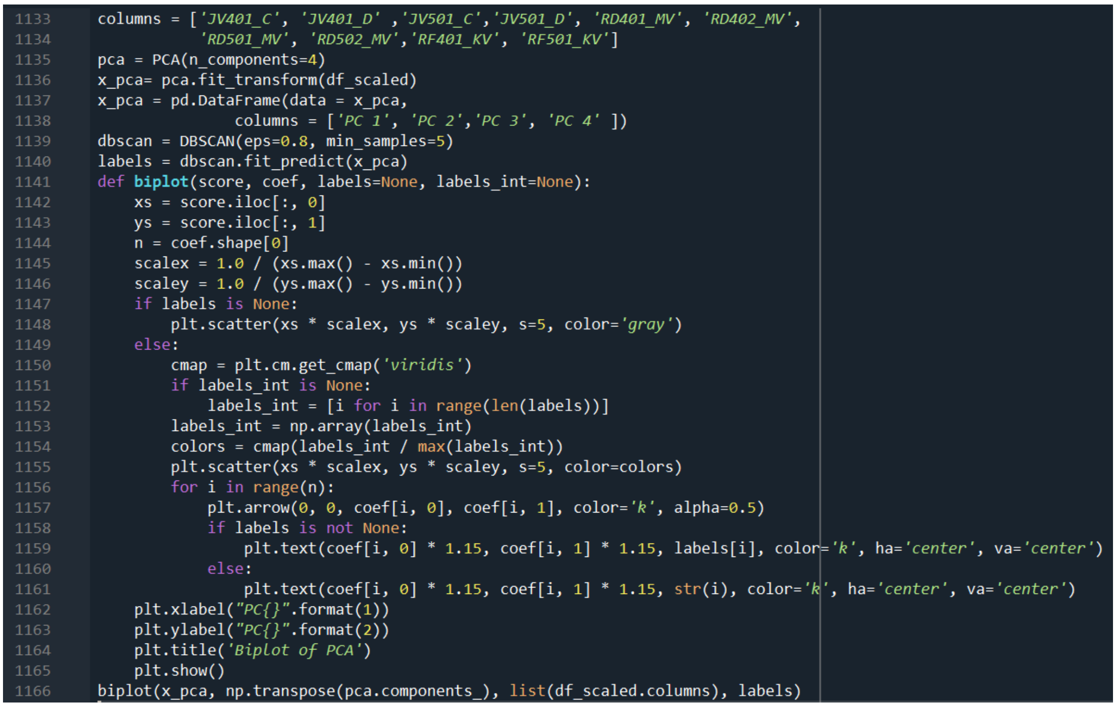
```python
import matplotlib.pyplot as plt

# Get the loadings matrix from PCA
loadings = pca.components_

# Plot feature vectors for each PC
plt.figure(figsize=(10, 6))
for i, pc in enumerate(loadings):
    plt.arrow(0, 0, pc[0], pc[1], color='r', alpha=0.5)
    plt.text(pc[0]*1.2, pc[1]*1.2, "Sensor {}".format(i+1), color='g',
             ha='center', va='center')
plt.xlim([-1, 1])
plt.ylim([-1, 1])
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.title("Feature vectors for each principal component")
plt.show()
```

A.7 Loadings with Principal Components



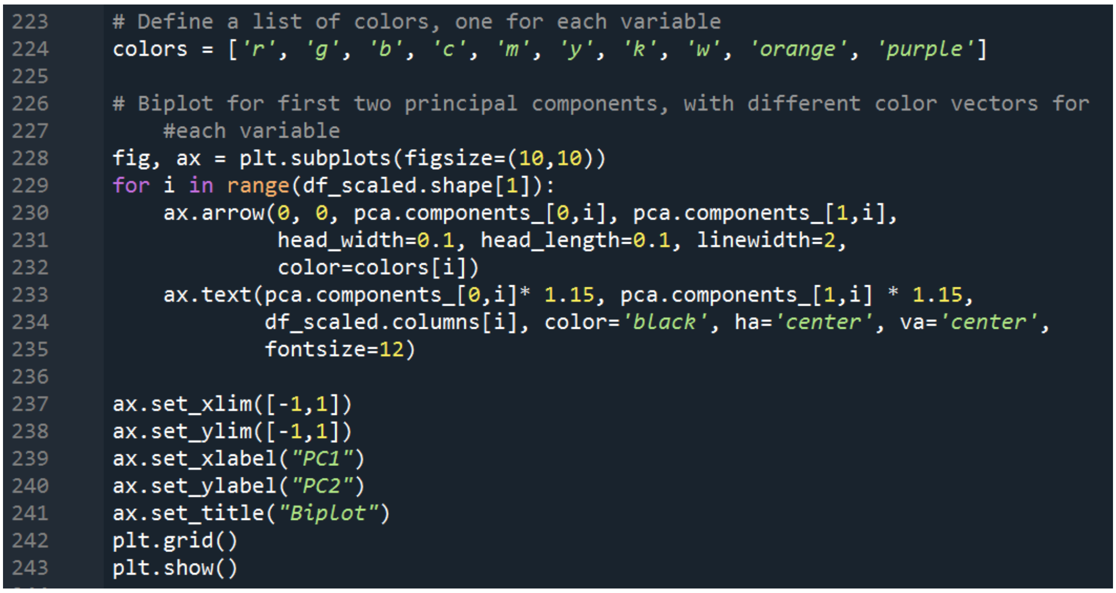
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
columns = df_scaled.columns.tolist()
loadings = pca.components_
# Create a plot with subplots for each principal component
fig, axs = plt.subplots(nrows=1, ncols=4, figsize=(15, 4))
# Loop over each principal component
for i in range(4):
    sorted_idxs = np.argsort(np.abs(loadings[i]))[::-1]
    sorted_loadings = loadings[i][sorted_idxs]
    sorted_columns = [columns[idx] for idx in sorted_idxs]
    # Create a bar plot of the columns and their loadings
    axs[i].bar(sorted_columns, sorted_loadings)
    axs[i].set_title(f'PC{i+1}')
    axs[i].set_xlabel('Column')
    axs[i].set_ylabel('Loading')
    axs[i].tick_params(axis='x', rotation=90)
# Hide the remaining subplots
for i in range(4, 10):
    axs[i].axis('off')
plt.tight_layout()
plt.show()
```

A.8 DBSCAN and Biplot



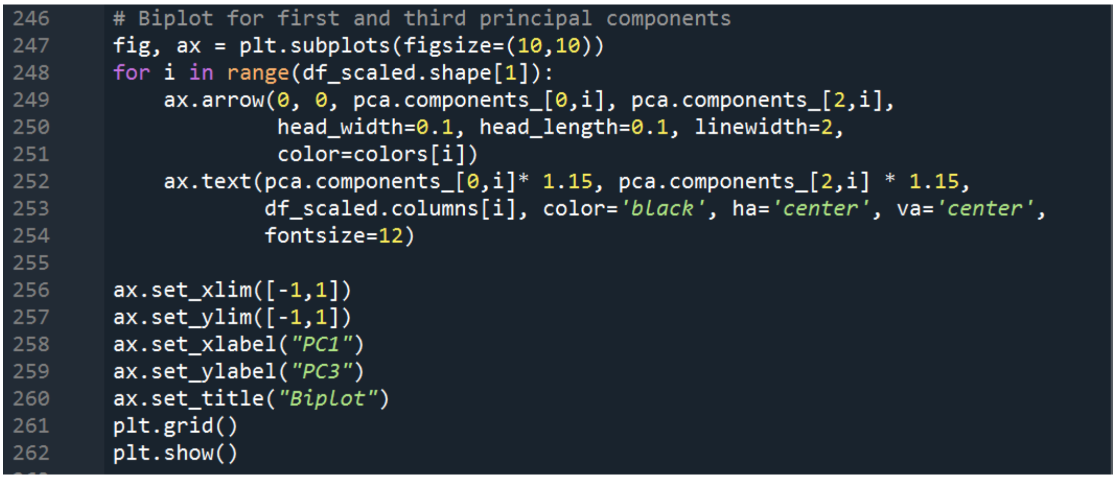
```python
columns = ['JV401_C', 'JV401_D' ,'JV501_C','JV501_D', 'RD401_MV', 'RD402_MV',
            'RD501_MV', 'RD502_MV','RF401_KV', 'RF501_KV']
pca = PCA(n_components=4)
x_pca= pca.fit_transform(df_scaled)
x_pca = pd.DataFrame(data = x_pca,
               columns = ['PC 1', 'PC 2','PC 3', 'PC 4' ])
dbscan = DBSCAN(eps=0.8, min_samples=5)
labels = dbscan.fit_predict(x_pca)
def biplot(score, coef, labels=None, labels_int=None):
    xs = score.iloc[:, 0]
    ys = score.iloc[:, 1]
    n = coef.shape[0]
    scalex = 1.0 / (xs.max() - xs.min())
    scaley = 1.0 / (ys.max() - ys.min())
    if labels is None:
        plt.scatter(xs * scalex, ys * scaley, s=5, color='gray')
    else:
        cmap = plt.cm.get_cmap('viridis')
        if labels_int is None:
            labels_int = [i for i in range(len(labels))]
        labels_int = np.array(labels_int)
        colors = cmap(labels_int / max(labels_int))
        plt.scatter(xs * scalex, ys * scaley, s=5, color=colors)
        for i in range(n):
            plt.arrow(0, 0, coef[i, 0], coef[i, 1], color='k', alpha=0.5)
            if labels is not None:
                plt.text(coef[i, 0] * 1.15, coef[i, 1] * 1.15, labels[i], color='k', ha='center', va='center')
            else:
                plt.text(coef[i, 0] * 1.15, coef[i, 1] * 1.15, str(i), color='k', ha='center', va='center')
    plt.xlabel("PC{}".format(1))
    plt.ylabel("PC{}".format(2))
    plt.title('Biplot of PCA')
    plt.show()
biplot(x_pca, np.transpose(pca.components_), list(df_scaled.columns), labels)
```

A.9 Biplot for PC1 and PC2



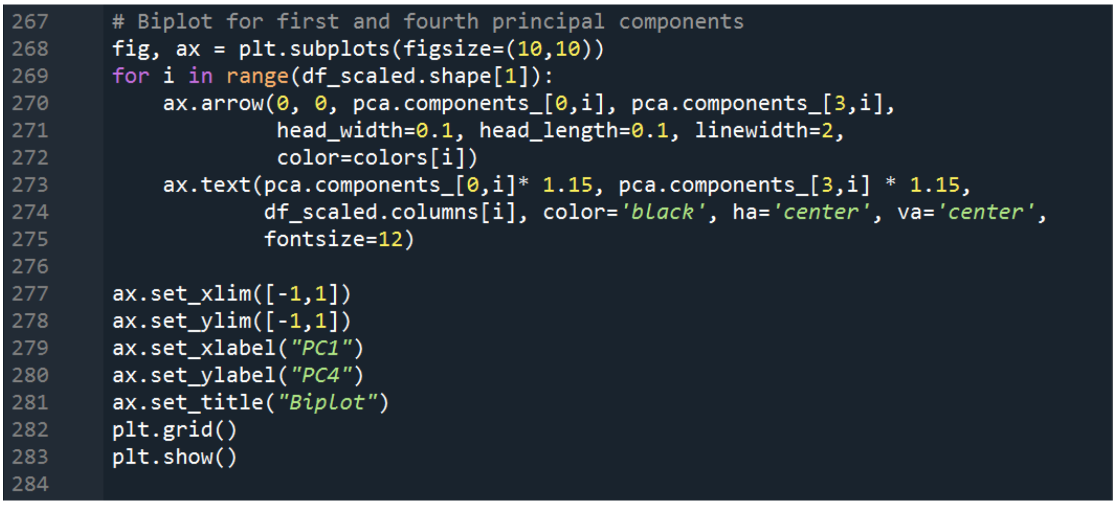
```python
# Define a list of colors, one for each variable
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k', 'w', 'orange', 'purple']

# Biplot for first two principal components, with different color vectors for
    #each variable
fig, ax = plt.subplots(figsize=(10,10))
for i in range(df_scaled.shape[1]):
    ax.arrow(0, 0, pca.components_[0,i], pca.components_[1,i],
             head_width=0.1, head_length=0.1, linewidth=2,
             color=colors[i])
    ax.text(pca.components_[0,i]* 1.15, pca.components_[1,i] * 1.15,
            df_scaled.columns[i], color='black', ha='center', va='center',
            fontsize=12)

ax.set_xlim([-1,1])
ax.set_ylim([-1,1])
ax.set_xlabel("PC1")
ax.set_ylabel("PC2")
ax.set_title("Biplot")
plt.grid()
plt.show()
```

A.10 Biplot for PC1 and PC3



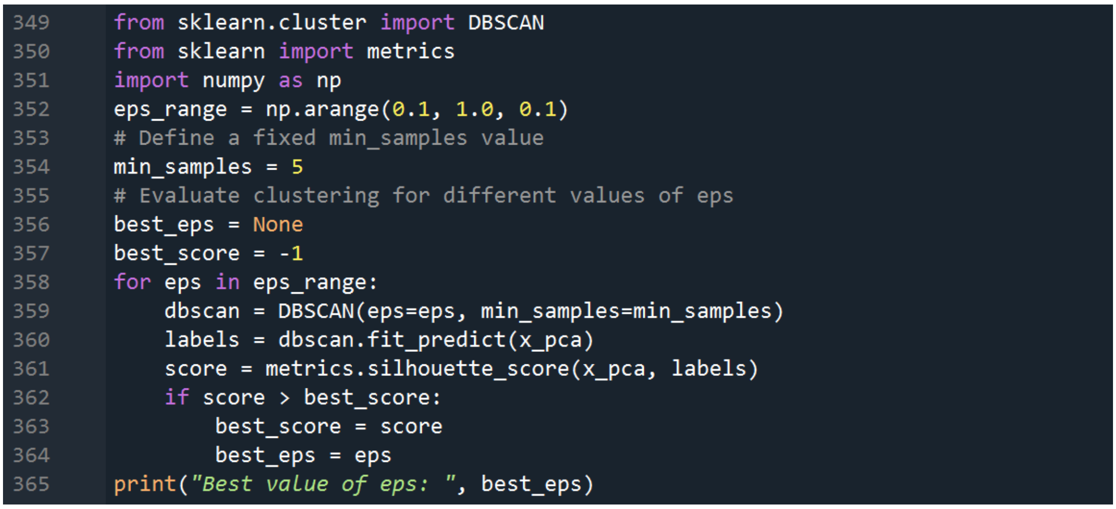
```python
# Biplot for first and third principal components
fig, ax = plt.subplots(figsize=(10,10))
for i in range(df_scaled.shape[1]):
    ax.arrow(0, 0, pca.components_[0,i], pca.components_[2,i],
             head_width=0.1, head_length=0.1, linewidth=2,
             color=colors[i])
    ax.text(pca.components_[0,i]* 1.15, pca.components_[2,i] * 1.15,
            df_scaled.columns[i], color='black', ha='center', va='center',
            fontsize=12)

ax.set_xlim([-1,1])
ax.set_ylim([-1,1])
ax.set_xlabel("PC1")
ax.set_ylabel("PC3")
ax.set_title("Biplot")
plt.grid()
plt.show()
```

A.11 Biplot for PC1 and PC4



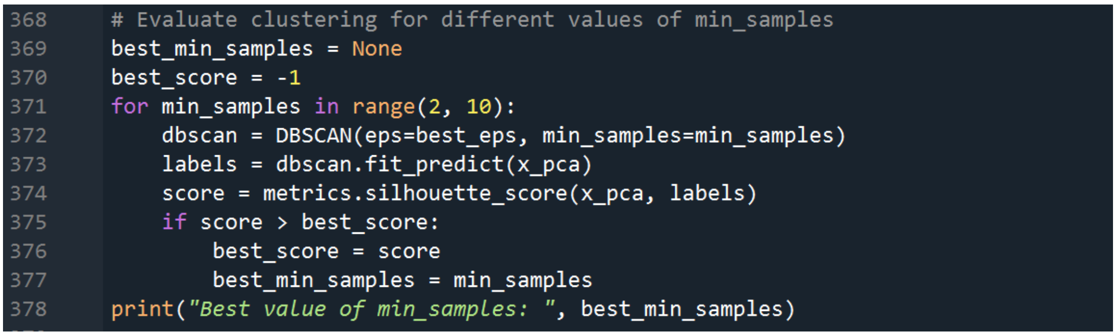
```python
# Biplot for first and fourth principal components
fig, ax = plt.subplots(figsize=(10,10))
for i in range(df_scaled.shape[1]):
    ax.arrow(0, 0, pca.components_[0,i], pca.components_[3,i],
             head_width=0.1, head_length=0.1, linewidth=2,
             color=colors[i])
    ax.text(pca.components_[0,i]* 1.15, pca.components_[3,i] * 1.15,
            df_scaled.columns[i], color='black', ha='center', va='center',
            fontsize=12)

ax.set_xlim([-1,1])
ax.set_ylim([-1,1])
ax.set_xlabel("PC1")
ax.set_ylabel("PC4")
ax.set_title("Biplot")
plt.grid()
plt.show()
```

A.12 DBSCAN



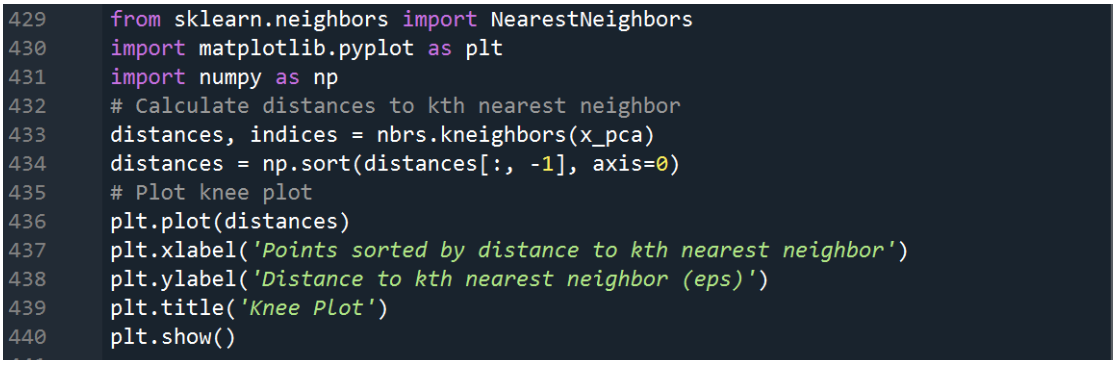
```python
from sklearn.cluster import DBSCAN
from sklearn import metrics
import numpy as np
eps_range = np.arange(0.1, 1.0, 0.1)
# Define a fixed min_samples value
min_samples = 5
# Evaluate clustering for different values of eps
best_eps = None
best_score = -1
for eps in eps_range:
    dbscan = DBSCAN(eps=eps, min_samples=min_samples)
    labels = dbscan.fit_predict(x_pca)
    score = metrics.silhouette_score(x_pca, labels)
    if score > best_score:
        best_score = score
        best_eps = eps
print("Best value of eps: ", best_eps)
```

A.13 Best Min-sample



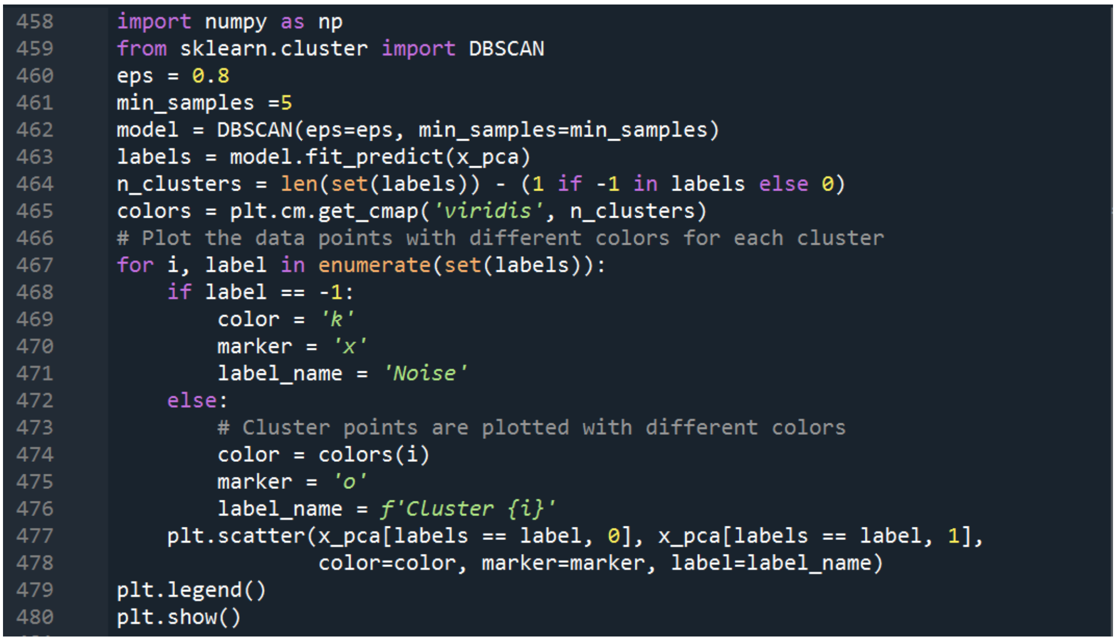
```python
# Evaluate clustering for different values of min_samples
best_min_samples = None
best_score = -1
for min_samples in range(2, 10):
    dbscan = DBSCAN(eps=best_eps, min_samples=min_samples)
    labels = dbscan.fit_predict(x_pca)
    score = metrics.silhouette_score(x_pca, labels)
    if score > best_score:
        best_score = score
        best_min_samples = min_samples
print("Best value of min_samples: ", best_min_samples)
```

A.14 Knee Plot



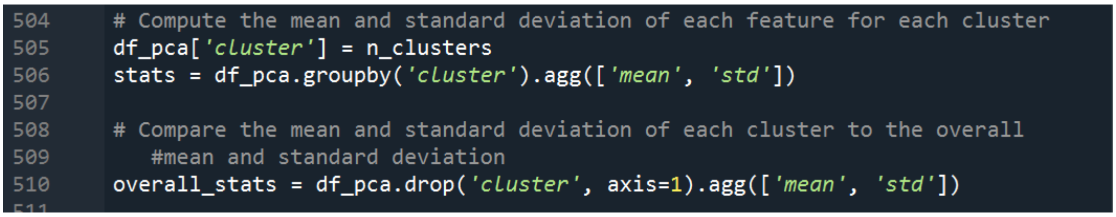
```python
from sklearn.neighbors import NearestNeighbors
import matplotlib.pyplot as plt
import numpy as np
# Calculate distances to kth nearest neighbor
distances, indices = nbrs.kneighbors(x_pca)
distances = np.sort(distances[:, -1], axis=0)
# Plot knee plot
plt.plot(distances)
plt.xlabel('Points sorted by distance to kth nearest neighbor')
plt.ylabel('Distance to kth nearest neighbor (eps)')
plt.title('Knee Plot')
plt.show()
```

A.15 The Clusters



```python
import numpy as np
from sklearn.cluster import DBSCAN
eps = 0.8
min_samples =5
model = DBSCAN(eps=eps, min_samples=min_samples)
labels = model.fit_predict(x_pca)
n_clusters = len(set(labels)) - (1 if -1 in labels else 0)
colors = plt.cm.get_cmap('viridis', n_clusters)
# Plot the data points with different colors for each cluster
for i, label in enumerate(set(labels)):
    if label == -1:
        color = 'k'
        marker = 'x'
        label_name = 'Noise'
    else:
        # Cluster points are plotted with different colors
        color = colors(i)
        marker = 'o'
        label_name = f'Cluster {i}'
    plt.scatter(x_pca[labels == label, 0], x_pca[labels == label, 1],
                color=color, marker=marker, label=label_name)
plt.legend()
plt.show()
```

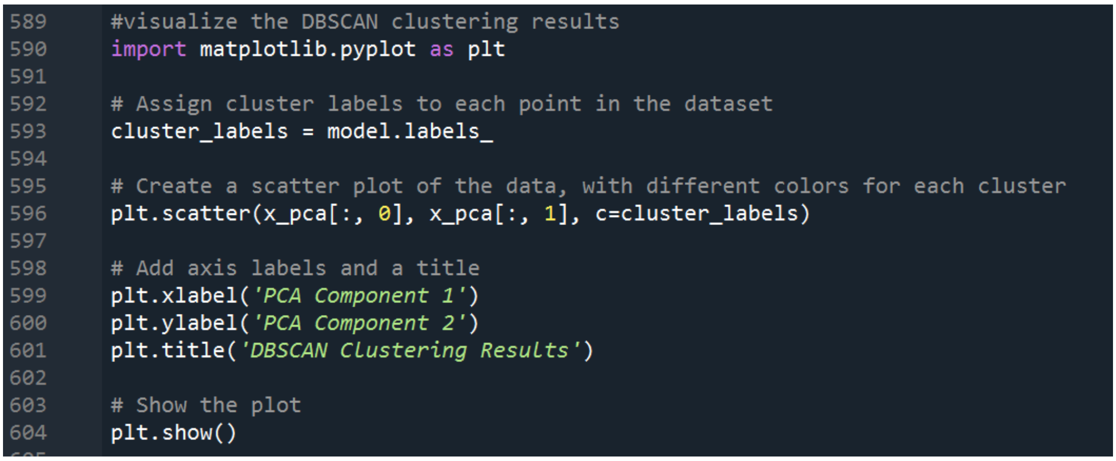
A.16 The Mean and Standard Deviation of Features in the Clusters



```python
# Compute the mean and standard deviation of each feature for each cluster
df_pca['cluster'] = n_clusters
stats = df_pca.groupby('cluster').agg(['mean', 'std'])

# Compare the mean and standard deviation of each cluster to the overall
   #mean and standard deviation
overall_stats = df_pca.drop('cluster', axis=1).agg(['mean', 'std'])
```

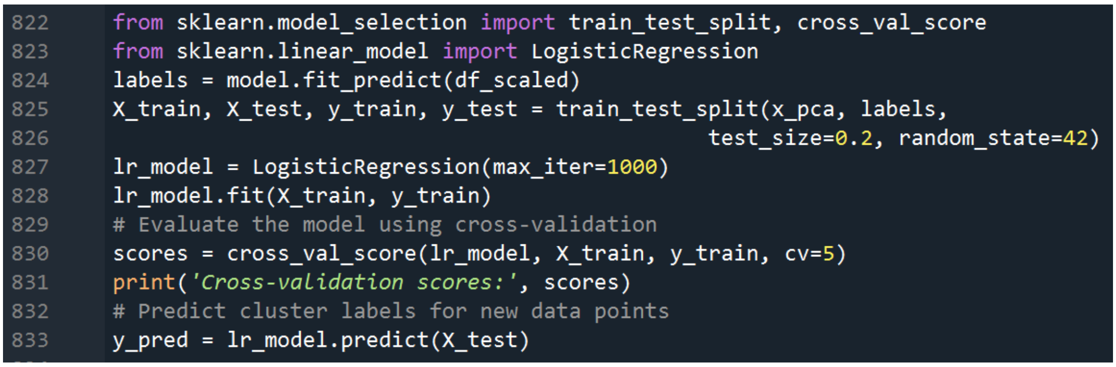
This type of analysis can help identify patterns and differences between clusters, providing a deeper understanding of how each feature contributes to the clustering results.

A.17 Visualize the DBSCAN clustering Results



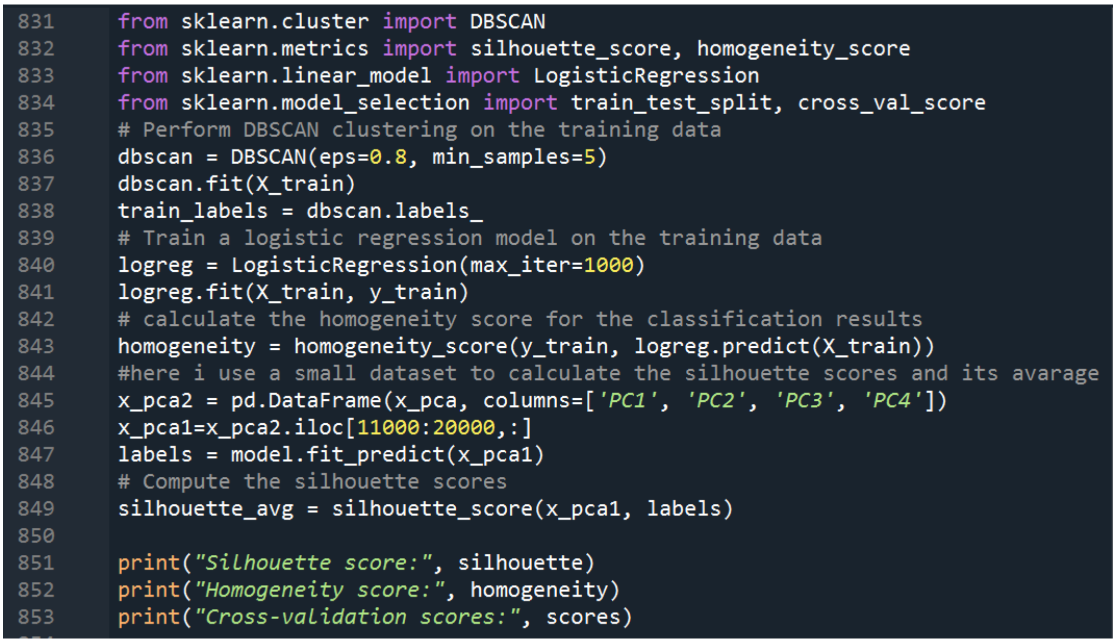
```
#visualize the DBSCAN clustering results
import matplotlib.pyplot as plt

# Assign cluster labels to each point in the dataset
cluster_labels = model.labels_

# Create a scatter plot of the data, with different colors for each cluster
plt.scatter(x_pca[:, 0], x_pca[:, 1], c=cluster_labels)

# Add axis labels and a title
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('DBSCAN Clustering Results')

# Show the plot
plt.show()
```

A.18 Cross Validation



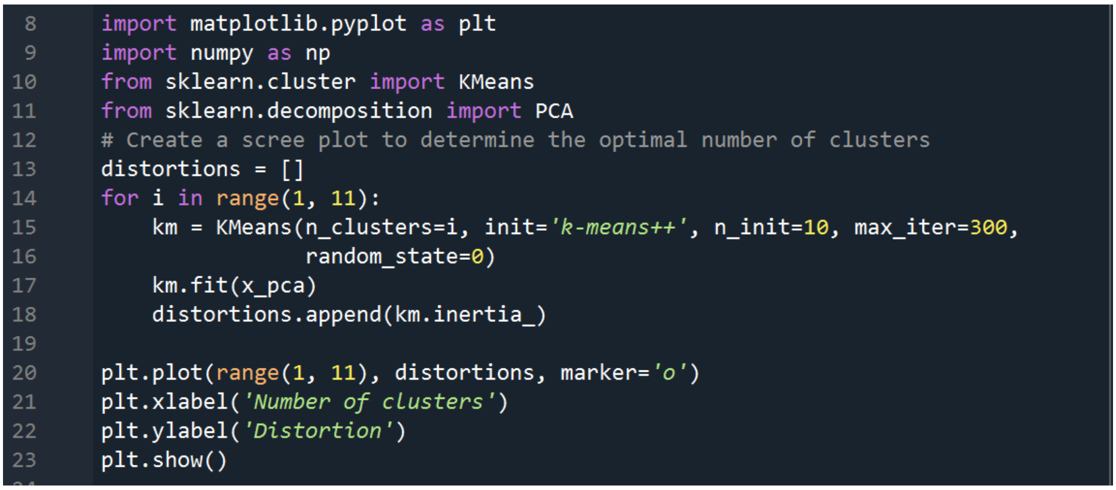
```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
labels = model.fit_predict(df_scaled)
X_train, X_test, y_train, y_test = train_test_split(x_pca, labels,
                                                test_size=0.2, random_state=42)
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)
# Evaluate the model using cross-validation
scores = cross_val_score(lr_model, X_train, y_train, cv=5)
print('Cross-validation scores:', scores)
# Predict cluster labels for new data points
y_pred = lr_model.predict(X_test)
```

A.19 Silhouette and Homogeneity



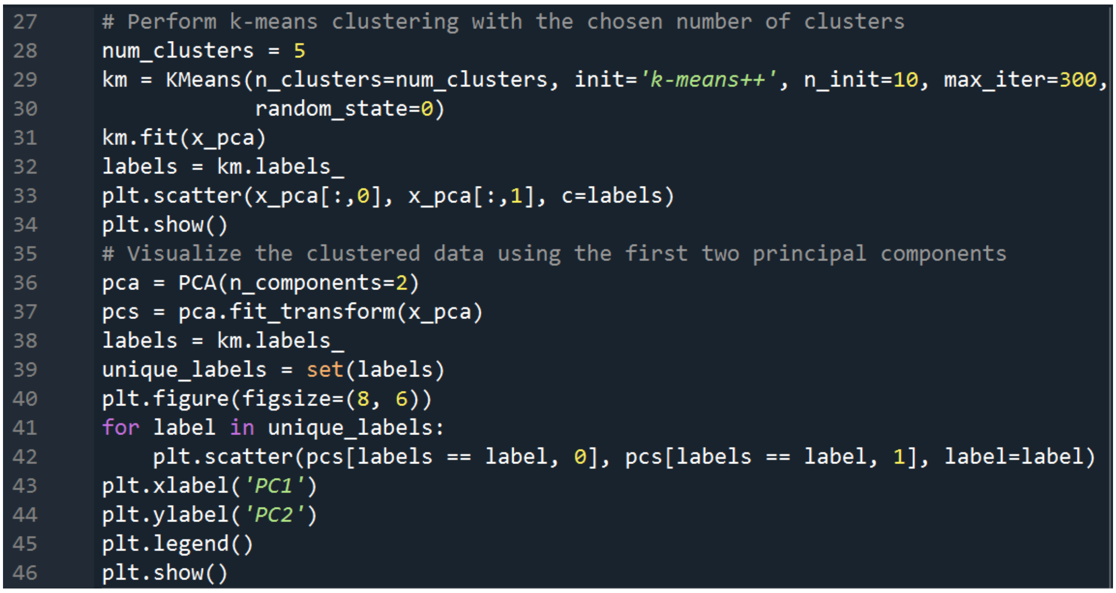
```
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score, homogeneity_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score
# Perform DBSCAN clustering on the training data
dbscan = DBSCAN(eps=0.8, min_samples=5)
dbscan.fit(X_train)
train_labels = dbscan.labels_
# Train a logistic regression model on the training data
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train, y_train)
# calculate the homogeneity score for the classification results
homogeneity = homogeneity_score(y_train, logreg.predict(X_train))
#here i use a small dataset to calculate the silhouette scores and its avarage
x_pca2 = pd.DataFrame(x_pca, columns=['PC1', 'PC2', 'PC3', 'PC4'])
x_pca1=x_pca2.iloc[11000:20000,:]
labels = model.fit_predict(x_pca1)
# Compute the silhouette scores
silhouette_avg = silhouette_score(x_pca1, labels)

print("Silhouette score:", silhouette)
print("Homogeneity score:", homogeneity)
print("Cross-validation scores:", scores)
```

A.20 K-Means Clustering



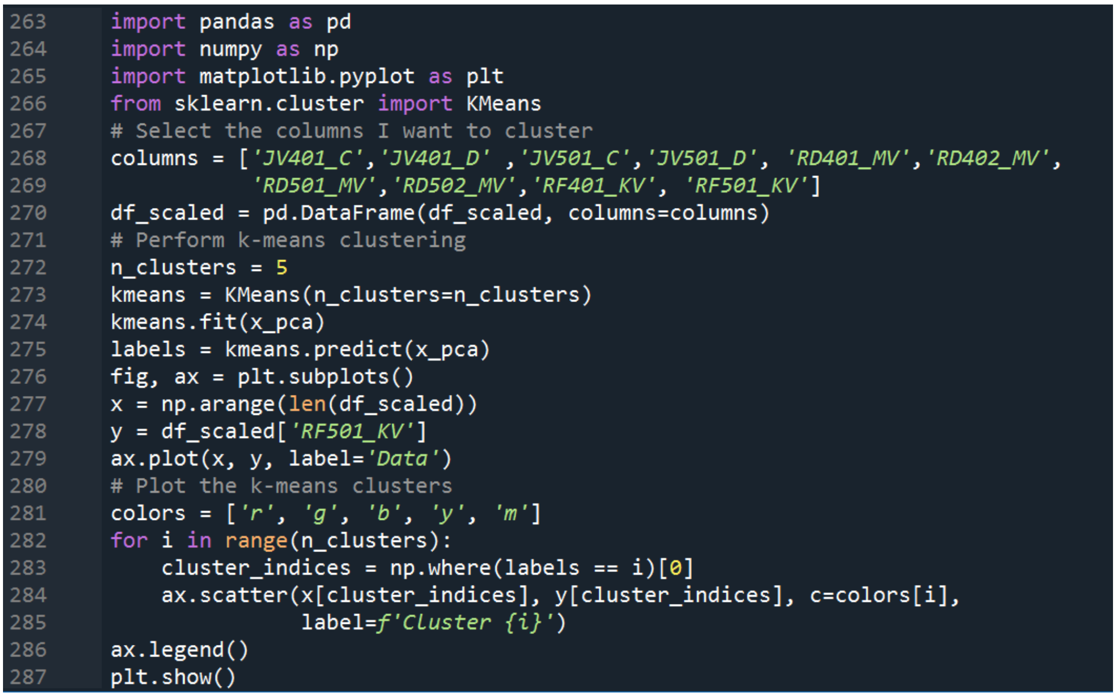
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
# Create a scree plot to determine the optimal number of clusters
distortions = []
for i in range(1, 11):
    km = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=300,
                random_state=0)
    km.fit(x_pca)
    distortions.append(km.inertia_)

plt.plot(range(1, 11), distortions, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.show()
```

A.21 Perform K-Means with PCs



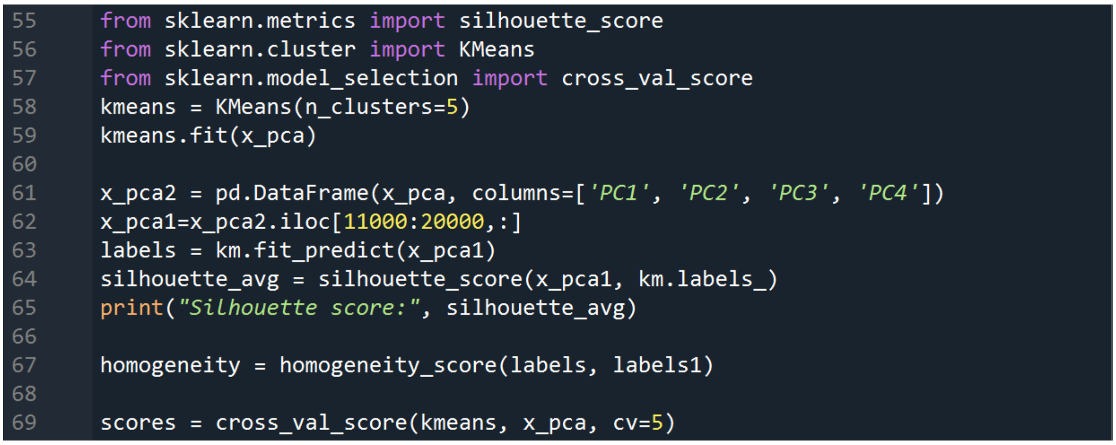
```python
# Perform k-means clustering with the chosen number of clusters
num_clusters = 5
km = KMeans(n_clusters=num_clusters, init='k-means++', n_init=10, max_iter=300,
            random_state=0)
km.fit(x_pca)
labels = km.labels_
plt.scatter(x_pca[:,0], x_pca[:,1], c=labels)
plt.show()
# Visualize the clustered data using the first two principal components
pca = PCA(n_components=2)
pcs = pca.fit_transform(x_pca)
labels = km.labels_
unique_labels = set(labels)
plt.figure(figsize=(8, 6))
for label in unique_labels:
    plt.scatter(pcs[labels == label, 0], pcs[labels == label, 1], label=label)
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.show()
```

A.22 Perform K-Means with 5 Clusters



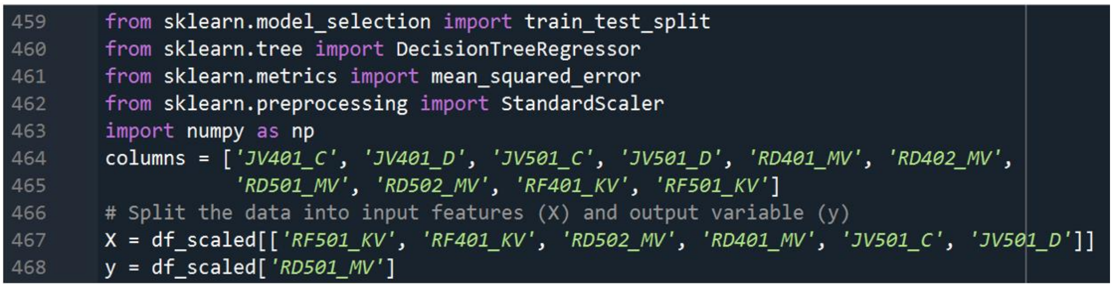
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
# Select the columns I want to cluster
columns = ['JV401_C','JV401_D' ,'JV501_C','JV501_D', 'RD401_MV','RD402_MV',
           'RD501_MV','RD502_MV','RF401_KV', 'RF501_KV']
df_scaled = pd.DataFrame(df_scaled, columns=columns)
# Perform k-means clustering
n_clusters = 5
kmeans = KMeans(n_clusters=n_clusters)
kmeans.fit(x_pca)
labels = kmeans.predict(x_pca)
fig, ax = plt.subplots()
x = np.arange(len(df_scaled))
y = df_scaled['RF501_KV']
ax.plot(x, y, label='Data')
# Plot the k-means clusters
colors = ['r', 'g', 'b', 'y', 'm']
for i in range(n_clusters):
    cluster_indices = np.where(labels == i)[0]
    ax.scatter(x[cluster_indices], y[cluster_indices], c=colors[i],
               label=f'Cluster {i}')
ax.legend()
plt.show()
```

A.23 Silhouette, Homogeneity and Cross Validation Scores for K-Means

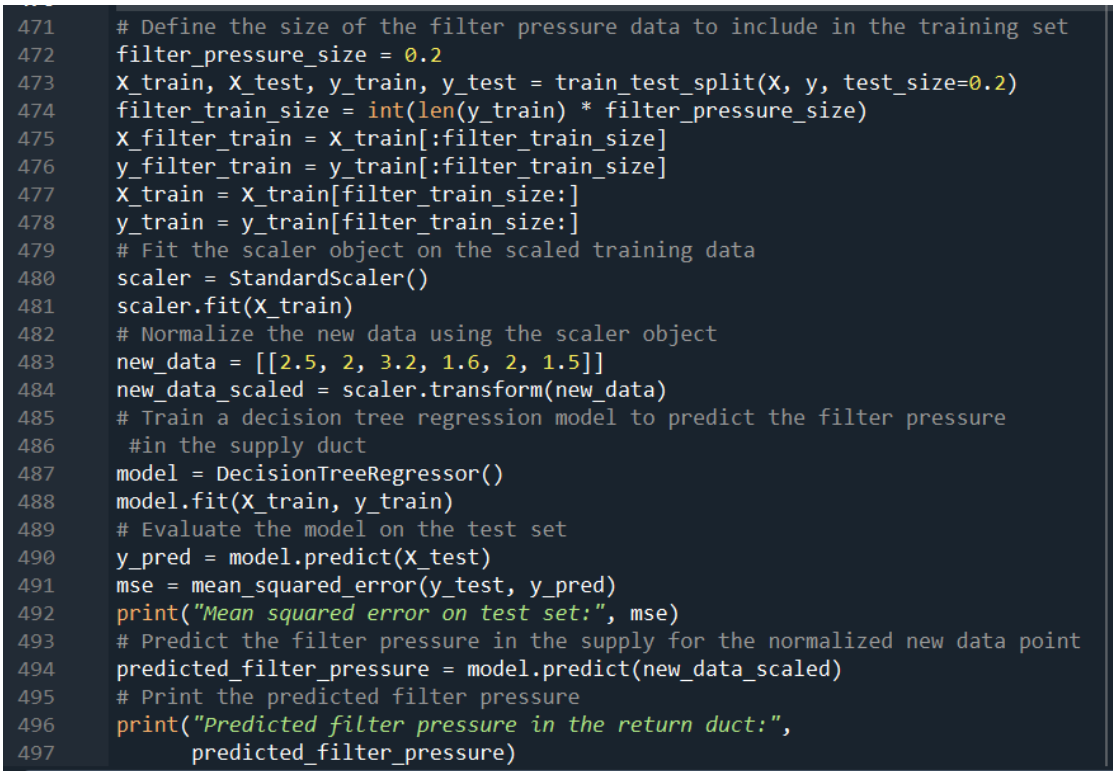


```
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans
from sklearn.model_selection import cross_val_score
kmeans = KMeans(n_clusters=5)
kmeans.fit(x_pca)

x_pca2 = pd.DataFrame(x_pca, columns=['PC1', 'PC2', 'PC3', 'PC4'])
x_pca1=x_pca2.iloc[11000:20000,:]
labels = km.fit_predict(x_pca1)
silhouette_avg = silhouette_score(x_pca1, km.labels_)
print("Silhouette score:", silhouette_avg)

homogeneity = homogeneity_score(labels, labels1)

scores = cross_val_score(kmeans, x_pca, cv=5)
```

A.24 DecisionTreeRegressor and Predictive Model

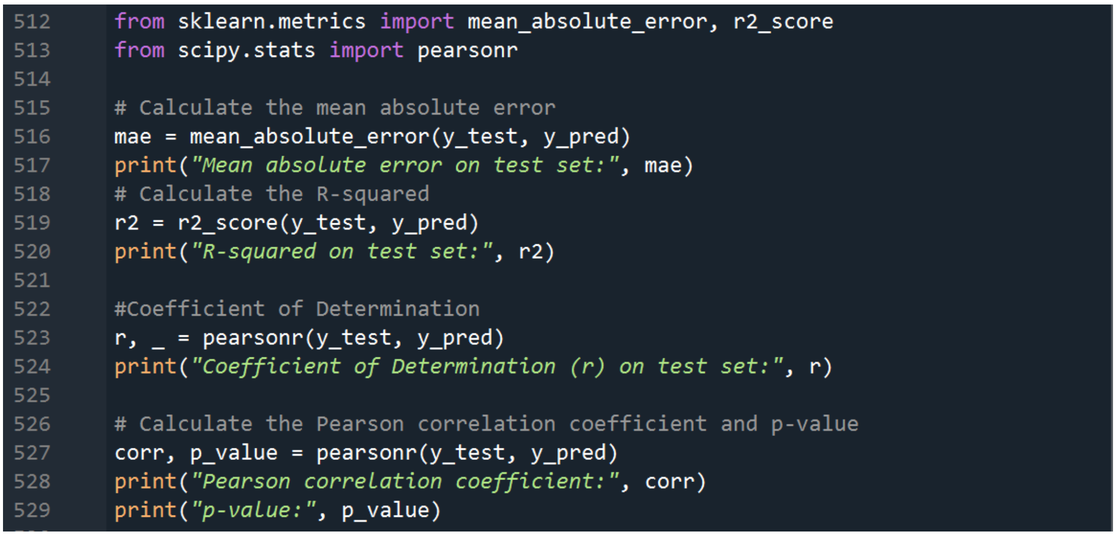


```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
import numpy as np
columns = ['JV401_C', 'JV401_D', 'JV501_C', 'JV501_D', 'RD401_MV', 'RD402_MV',
           'RD501_MV', 'RD502_MV', 'RF401_KV', 'RF501_KV']
# Split the data into input features (X) and output variable (y)
X = df_scaled[['RF501_KV', 'RF401_KV', 'RD502_MV', 'RD401_MV', 'JV501_C', 'JV501_D']]
y = df_scaled['RD501_MV']
```



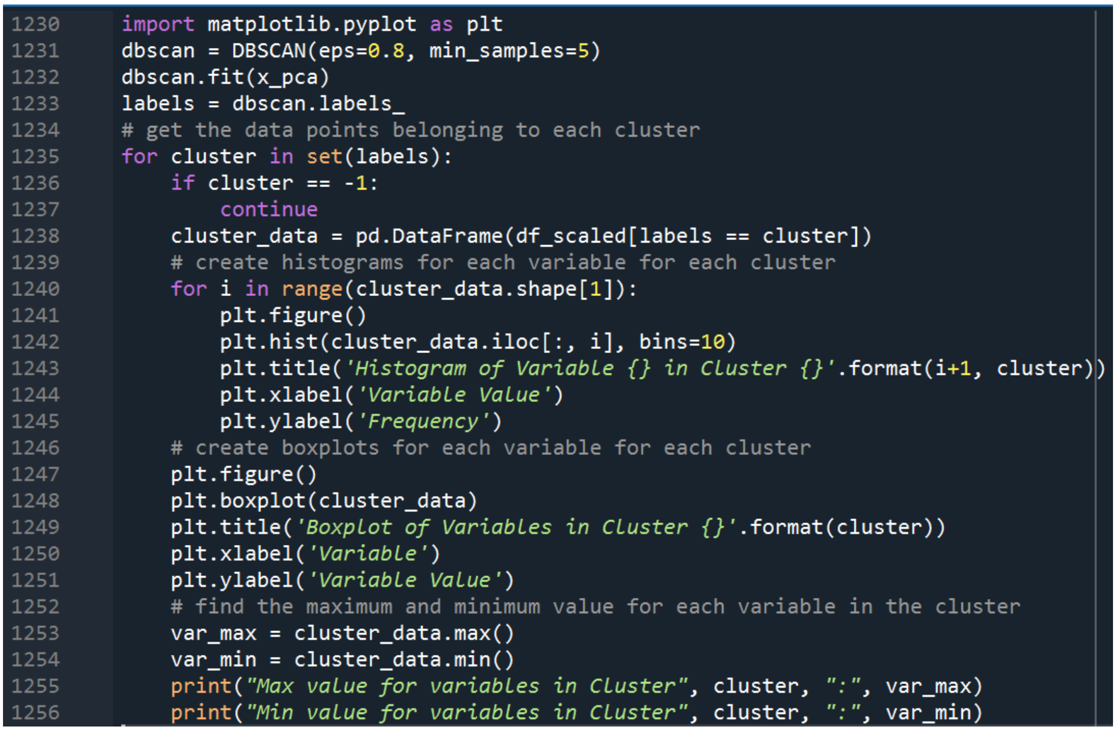
```
# Define the size of the filter pressure data to include in the training set
filter_pressure_size = 0.2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
filter_train_size = int(len(y_train) * filter_pressure_size)
X_filter_train = X_train[:filter_train_size]
y_filter_train = y_train[:filter_train_size]
X_train = X_train[filter_train_size:]
y_train = y_train[filter_train_size:]
# Fit the scaler object on the scaled training data
scaler = StandardScaler()
scaler.fit(X_train)
# Normalize the new data using the scaler object
new_data = [[2.5, 2, 3.2, 1.6, 2, 1.5]]
new_data_scaled = scaler.transform(new_data)
# Train a decision tree regression model to predict the filter pressure
 #in the supply duct
model = DecisionTreeRegressor()
model.fit(X_train, y_train)
# Evaluate the model on the test set
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print("Mean squared error on test set:", mse)
# Predict the filter pressure in the supply for the normalized new data point
predicted_filter_pressure = model.predict(new_data_scaled)
# Print the predicted filter pressure
print("Predicted filter pressure in the return duct:",
      predicted_filter_pressure)
```

A.25 Mean-absolute-error, R-squared, r, correlation Coefficient and P-value



```python
from sklearn.metrics import mean_absolute_error, r2_score
from scipy.stats import pearsonr

# Calculate the mean absolute error
mae = mean_absolute_error(y_test, y_pred)
print("Mean absolute error on test set:", mae)
# Calculate the R-squared
r2 = r2_score(y_test, y_pred)
print("R-squared on test set:", r2)

#Coefficient of Determination
r, _ = pearsonr(y_test, y_pred)
print("Coefficient of Determination (r) on test set:", r)

# Calculate the Pearson correlation coefficient and p-value
corr, p_value = pearsonr(y_test, y_pred)
print("Pearson correlation coefficient:", corr)
print("p-value:", p_value)
```

A.26 Maximum and Minimum Values



```python
import matplotlib.pyplot as plt
dbscan = DBSCAN(eps=0.8, min_samples=5)
dbscan.fit(x_pca)
labels = dbscan.labels_
# get the data points belonging to each cluster
for cluster in set(labels):
    if cluster == -1:
        continue
    cluster_data = pd.DataFrame(df_scaled[labels == cluster])
    # create histograms for each variable for each cluster
    for i in range(cluster_data.shape[1]):
        plt.figure()
        plt.hist(cluster_data.iloc[:, i], bins=10)
        plt.title('Histogram of Variable {} in Cluster {}'.format(i+1, cluster))
        plt.xlabel('Variable Value')
        plt.ylabel('Frequency')
    # create boxplots for each variable for each cluster
    plt.figure()
    plt.boxplot(cluster_data)
    plt.title('Boxplot of Variables in Cluster {}'.format(cluster))
    plt.xlabel('Variable')
    plt.ylabel('Variable Value')
    # find the maximum and minimum value for each variable in the cluster
    var_max = cluster_data.max()
    var_min = cluster_data.min()
    print("Max value for variables in Cluster", cluster, ":", var_max)
    print("Min value for variables in Cluster", cluster, ":", var_min)
```

NTNU

Norwegian University of
Science and Technology