Knowledge Discovery - Final Assignment

Stefan Neacsu

June 2025

1 Dataset Description: UCI Air Quality

The *UCI Air Quality* dataset comprises 9,358 hourly measurements recorded between March 2004 and February 2005 in an urban, traffic-influenced area of a mid-sized Italian city. The dataset was compiled by De Vito et al. (2008) and is available via the UCI Machine Learning Repository.[1]

The dataset includes:

- **Timestamp:** Date (DD/MM/YYYY) and Time (HH.MM.SS)
- **Pollutants (ground truth):**
  - CO (mg/m^3), NMHC and Benzene (µg/m^3), NO_x (ppb), NO_2 (µg/m^3)
- **Sensor responses:** Hourly averages from five metal-oxide sensors (PT08.S1–PT08.S5), each nominally targeted at CO, NMHC, NO_x, NO_2, and O_3, respectively.
- **Environmental variables:** Ambient temperature (°C), relative humidity (%), and absolute humidity.
- **Missing values:** Marked as `-200`, affecting pollutant and sensor readings.

This multivariate time-series dataset includes both discrete and continuous variables. It is known for exhibiting cross-sensitivities between sensors and long-term sensor drift. It represents one of the longest publicly available field deployments of low-cost chemical sensor arrays for air quality monitoring.

Applications. Common use cases include regression and time-series forecasting of pollutant concentrations, drift compensation modeling, and the calibration of low-cost sensors under varying environmental conditions.

2 Processing the dataset

2.1 Cleanup

The dataset cleanup step involves getting rid of NaN values, residing in two unnamed columns and a few on the other rows. Moreover, the date and time are unified into a single timestamp column.

Then, plots investigation is performed. Figures 1 and 2 display the evolution of pollutants and sensor measurements over time. The spikes to the bottom of the graph are indicators of missing data. Also, NMHC measurements exists only for a brief period of time, thus this column can be dropped. The boxplots (Figure 3) confirm the missing data as outliers, except for NMHC, where the missing values are so dominant that the actual measured valued are outliers themselves.

[1] `https://archive.ics.uci.edu/dataset/360/air+quality`

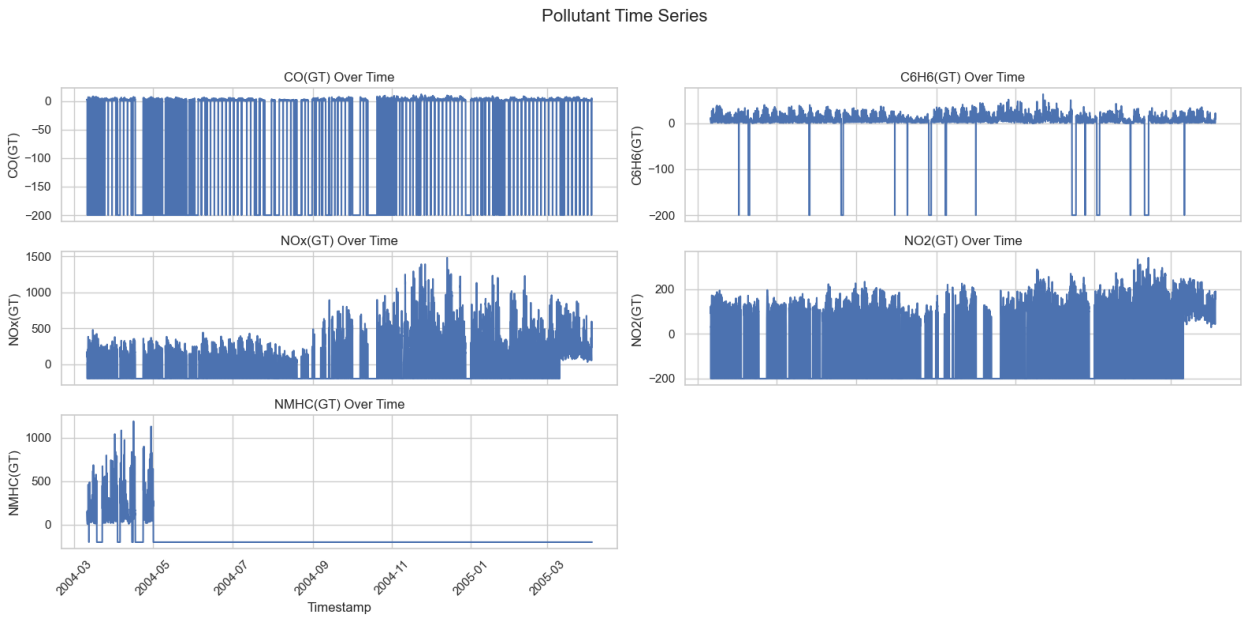


Figure 1: Pollutants time series

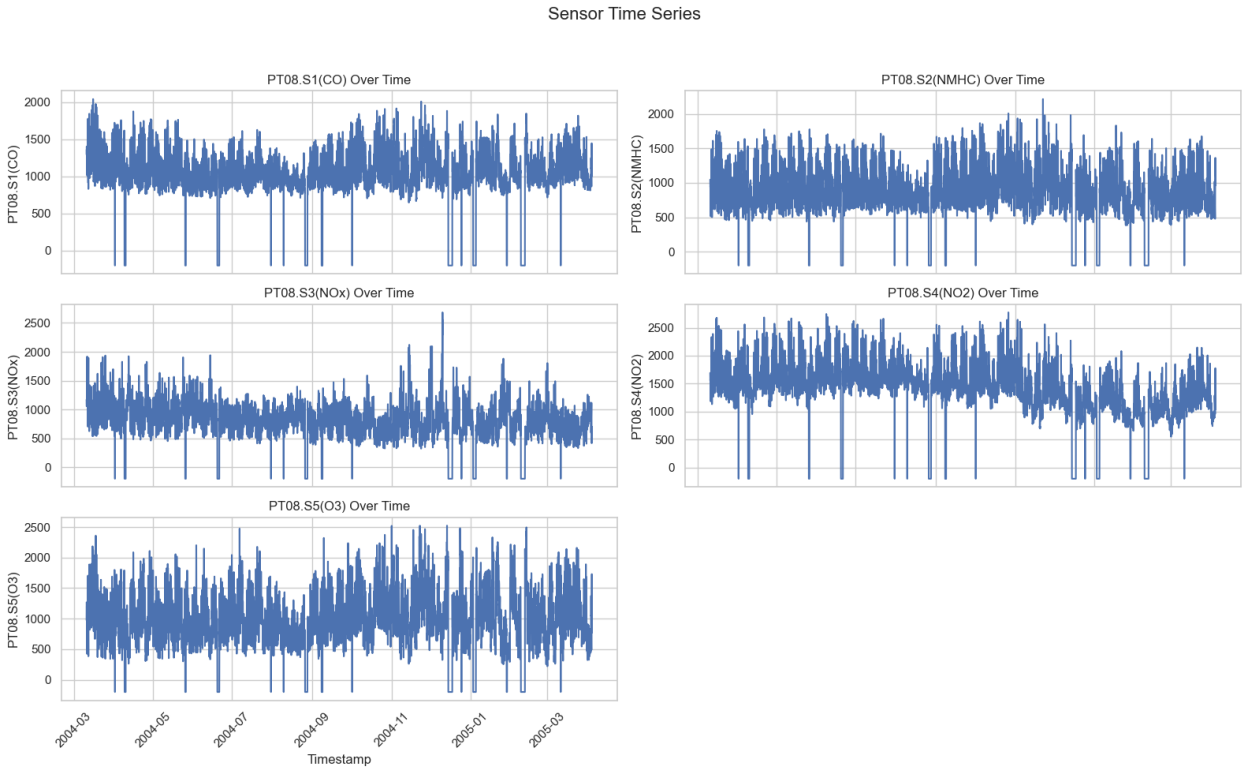


Figure 2: Pollutants time series

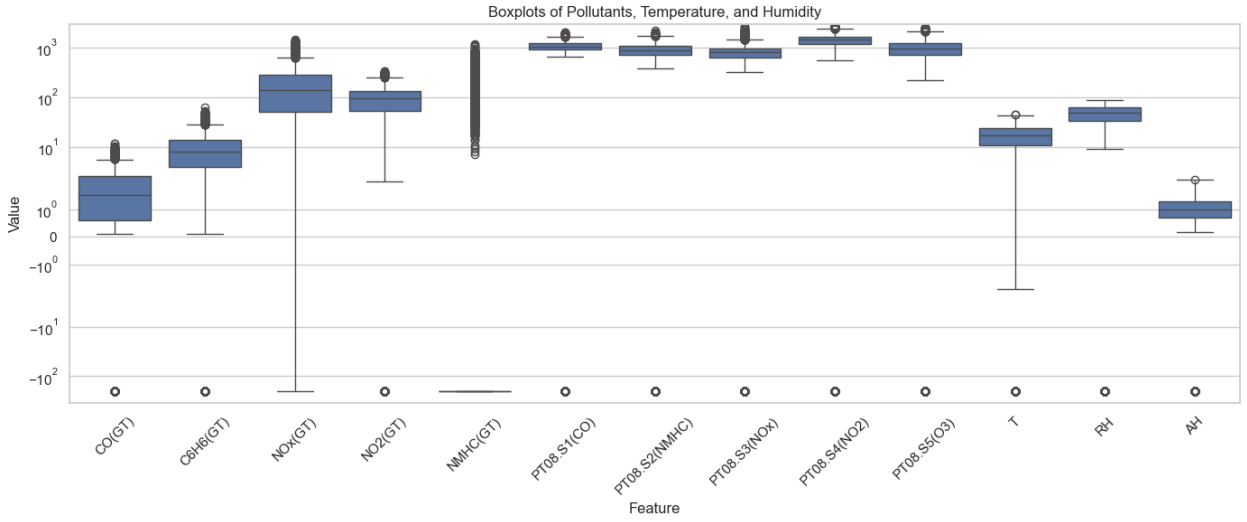


Figure 3: Box plots

The state of the data after cleanup is reflected in figures 4, 5, boxplot 6 and distribution plots 7, which show approximately normal or skew normal distributions.

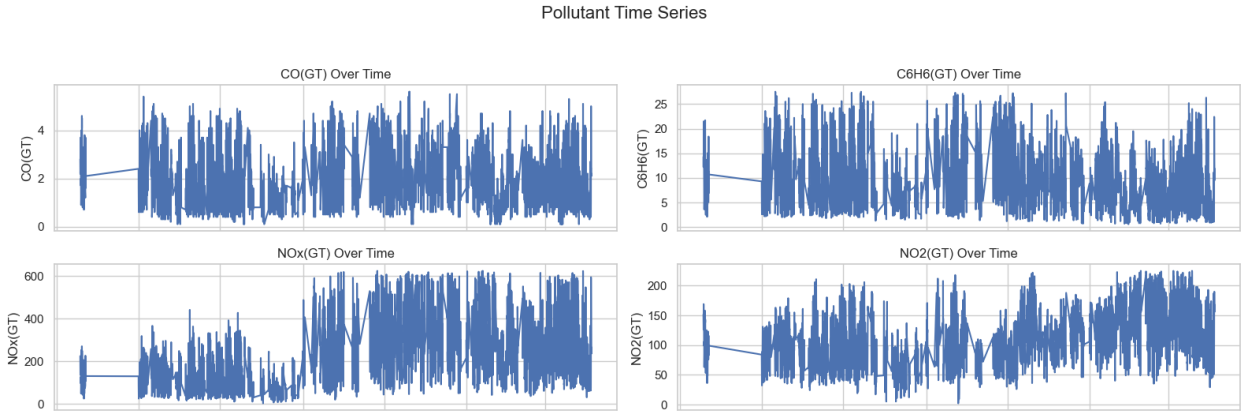


Figure 4: Pollution time series (cleaned)

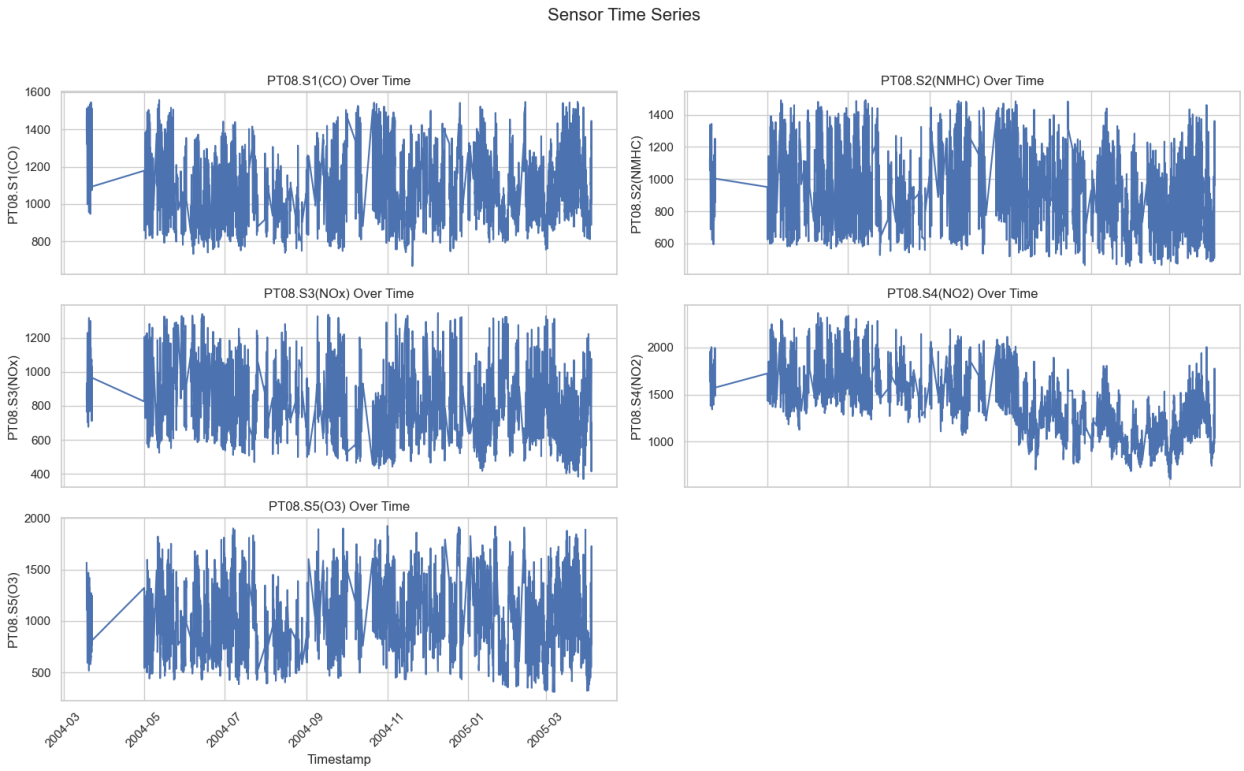


Figure 5: Sensors time series (cleaned)

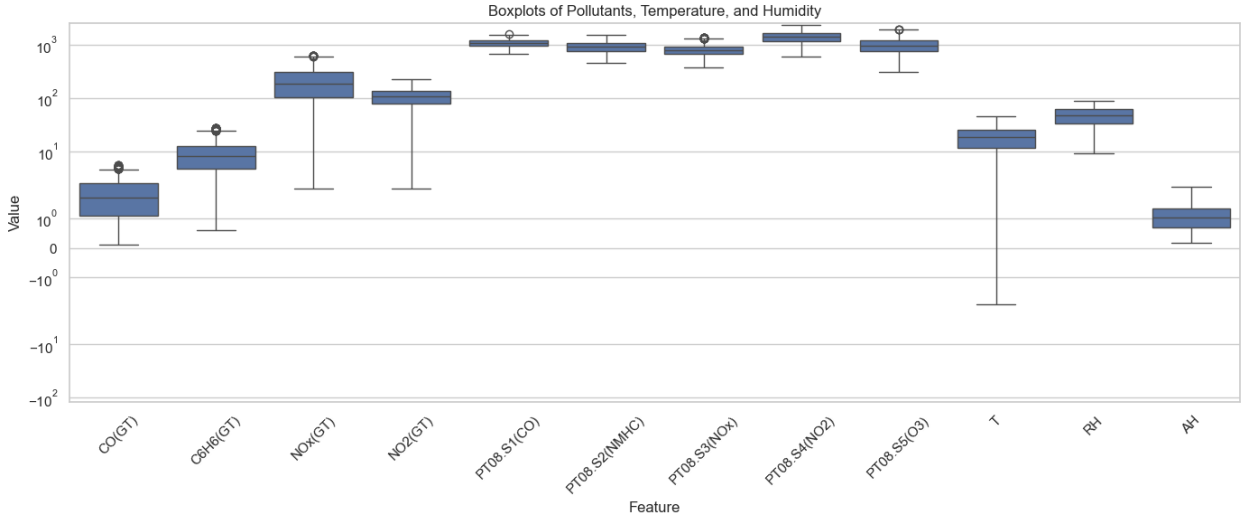


Figure 6: Box plots (cleaned)

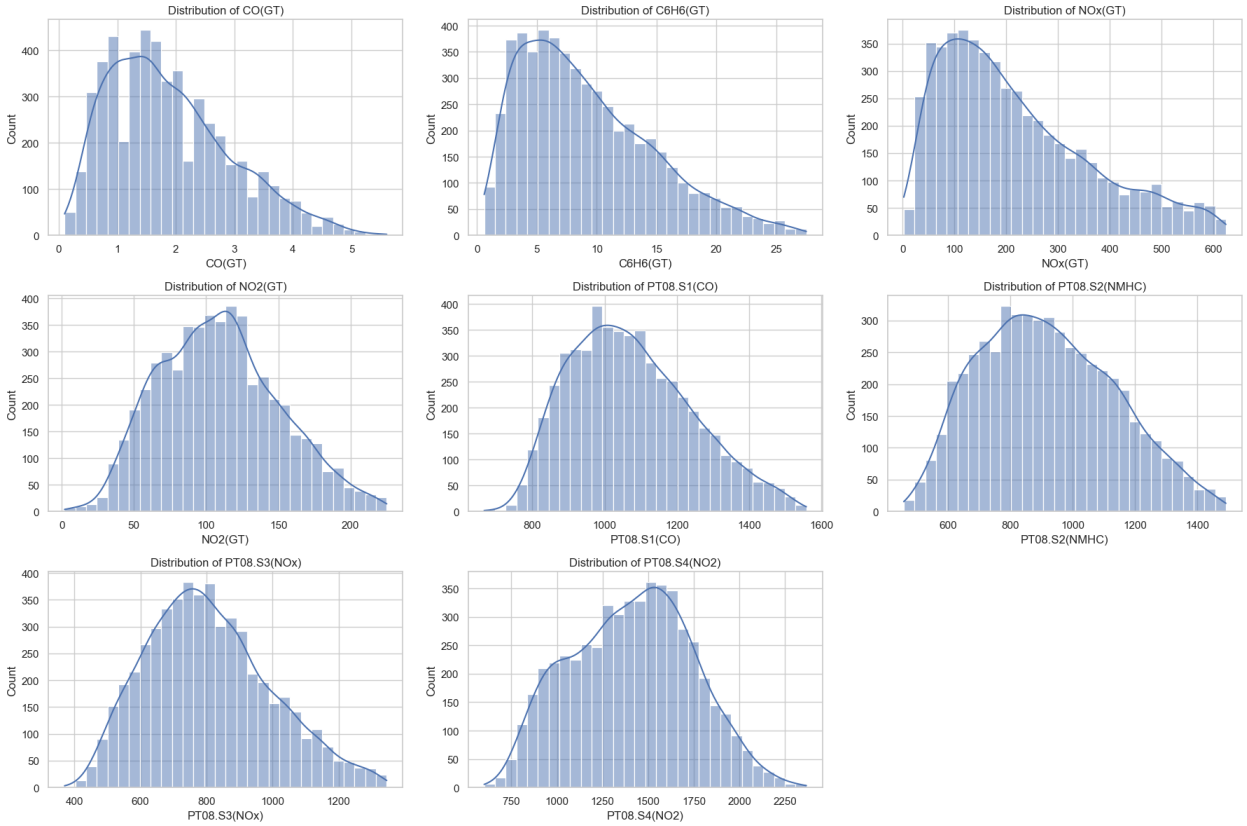


Figure 7: Distribution plots (cleaned)

The cleanup step leaves the dataset with 5335 samples with the features distributions explained in Table 1.

Variable	**Mean**	**Min**	**25%**	**50%**	**75%**	**Max**
Timestamp	2004-10-16	2004-03-18	2004-07-09	2004-10-24	2005-01-21	2005-04-04
CO(GT)	1.882043	0.100000	1.100000	1.700000	2.500000	5.600000
PT08.S1(CO)	1073.632802	667.000000	944.000000	1055.000000	1186.000000	1558.000000
C6H6(GT)	9.209709	0.600000	4.800000	8.100000	12.700000	27.500000
PT08.S2(NMHC)	918.943768	459.000000	753.000000	902.000000	1074.000000	1492.000000
NOx(GT)	217.773758	2.000000	104.000000	183.000000	305.000000	624.000000
PT08.S3(NOx)	807.407498	370.000000	669.000000	789.000000	925.500000	1346.000000
NO2(GT)	109.319213	2.000000	78.000000	108.000000	137.000000	225.000000
PT08.S4(NO2)	1408.425679	601.000000	1156.000000	1421.000000	1647.500000	2367.000000
PT08.S5(O3)	991.824742	310.000000	747.000000	967.000000	1215.000000	1925.000000
T	18.725361	-1.900000	11.800000	18.500000	25.200000	44.600000
RH	47.811771	9.200000	33.800000	47.800000	61.500000	88.700000
AH	1.023867	0.184700	0.704300	1.015100	1.321450	2.180600

Table 1: Descriptive statistics of sensor and pollution data

2.2 Data analysis

Principal Component Analysis (PCA) with 6 components was performed on the cleaned dataset, the first two components (which make up for about 80% of the explained variance) being scattered in Figure 8. It

can be observed that there exist a sort of linear separtion between pollutants like C6H6 and NO2, but there is no clear cluster structure.

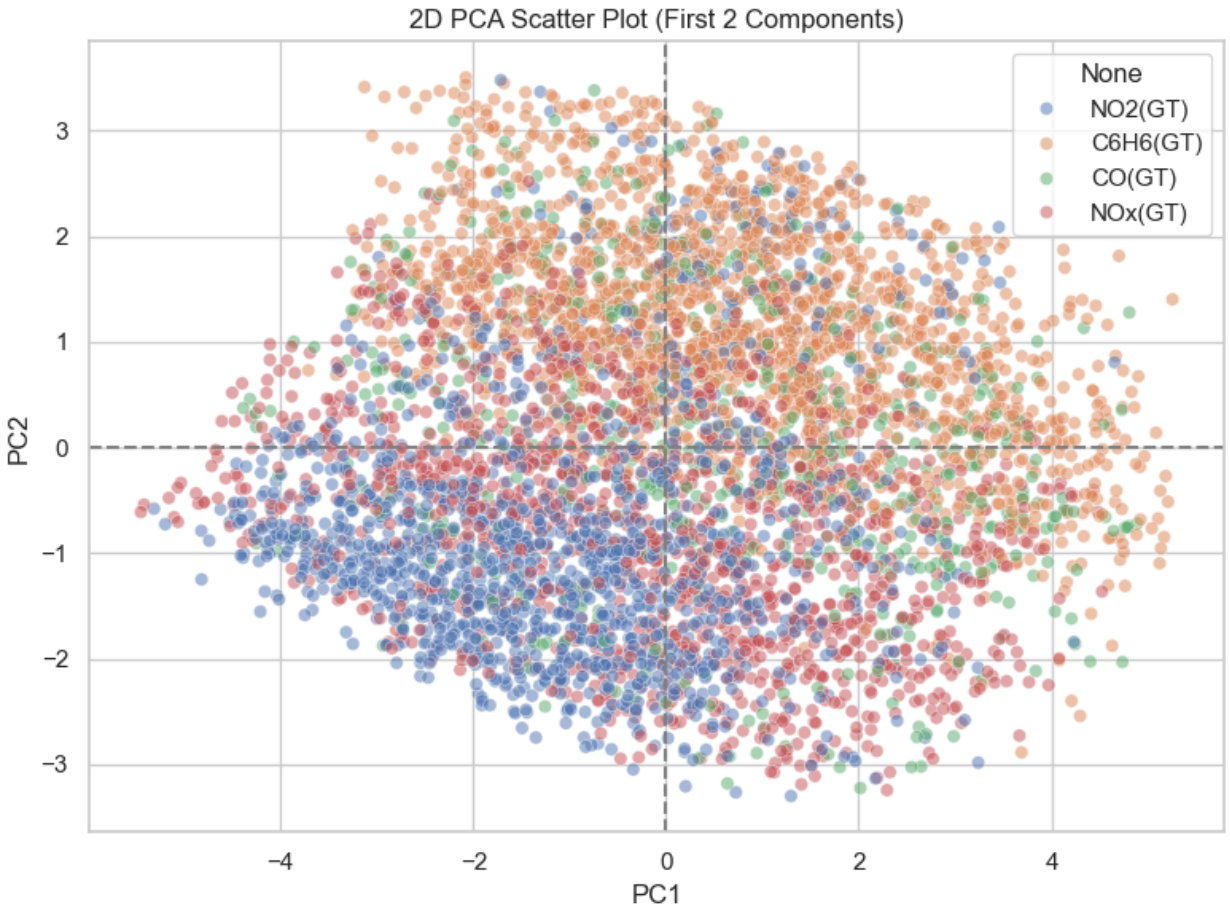


Figure 8: Plot of the first two PCA components

Next, the weights of a linear regression fit algorithm were used to observe the contribution of each sensor to detecting the presence of a certain chemical. The bar plot results are showcased in 9 and can open a wato interpretation.

2.2.1 CO(GT) (Carbon Monoxide)

- **PT08.S2(NMHC)** has the strongest positive weight, suggesting it's the most informative sensor for predicting CO levels.
- **PT08.S1(CO)**, which is expected to directly sense CO, also contributes positively — but less than PT08.S2.
- **PT08.S4(NO2)** has a weak negative impact, indicating a mild inverse relationship.

Conclusion: Surprisingly, **PT08.S2(NMHC)** (which detects volatile hydrocarbons) is more predictive of CO levels than the direct CO sensor. This could imply sensor crosstalk or shared sources of pollutants.

2.2.2 C6H6(GT) (Benzene)

- **PT08.S2(NMHC)** has an overwhelmingly dominant positive coefficient — far larger than all others.

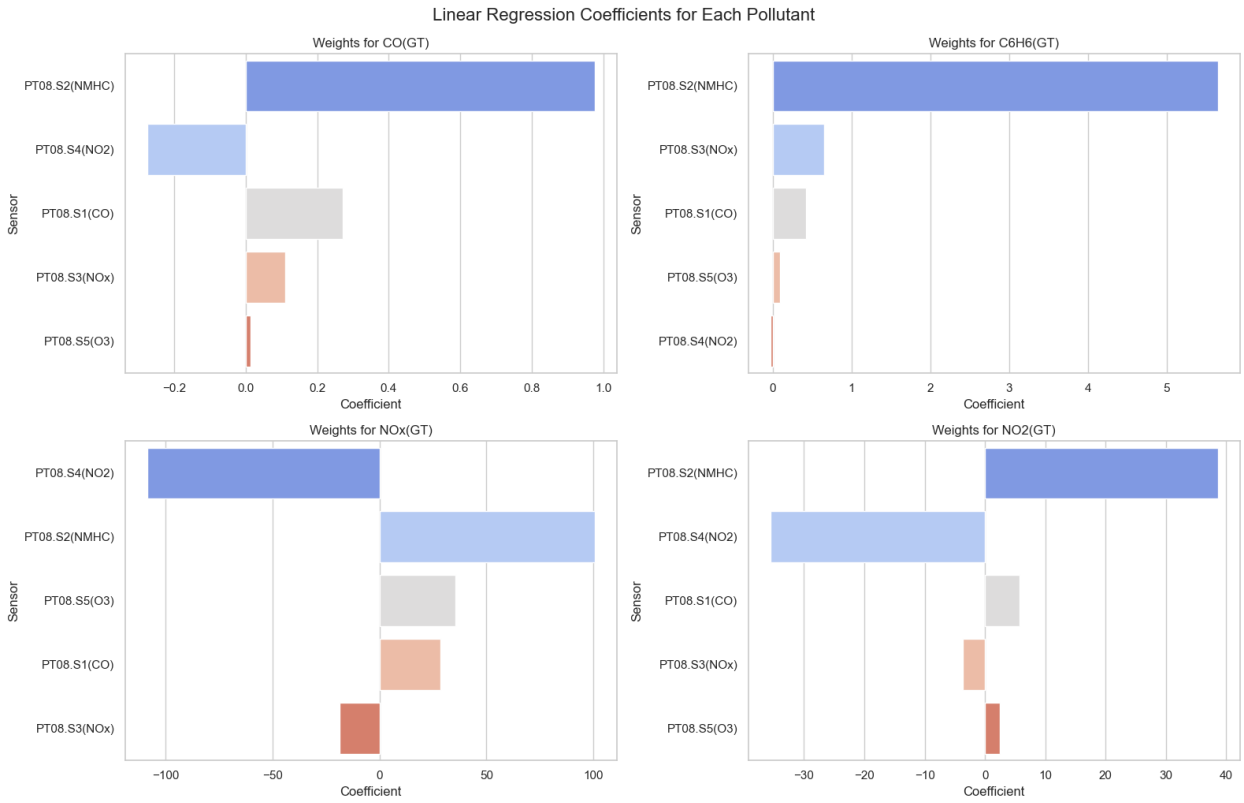


Figure 9: Linear regression weights of features per class

- Other sensors contribute negligibly.

Conclusion: The benzene level is almost exclusively predicted by **PT08.S2**, which aligns logically since this sensor is designed to detect non-methane hydrocarbons (which includes benzene). Very strong relationship.

2.2.3 NOx(GT) (Nitrogen Oxides)

- **PT08.S4(NO2)** and **PT08.S2(NMHC)** both have large positive contributions.
- **PT08.S1(CO)** and **PT08.S3(NOx)** have negative weights — counterintuitive for PT08.S3, which is supposed to detect NOx.

Conclusion: NOx levels are better explained by indirect sensors (**NO2** and **NMHC**) than the direct NOx sensor. This may suggest that the NOx sensor is noisy or nonlinear.

2.2.4 NO2(GT) (Nitrogen Dioxide)

- **PT08.S2(NMHC)** and **PT08.S4(NO2)** again dominate, with strong positive coefficients.
- **PT08.S3(NOx)** and **PT08.S5(O3)** have negative or weak effects.

Conclusion: Similar to NOx, **NO2** is best predicted by **PT08.S2** and **PT08.S4**, indicating potential shared chemical behavior or environmental correlation (e.g., traffic, industrial emissions).

3 ToscanaJ

3.1 Dataset discretization

The values in the cleaned dataset are still in continuous form. It needs to be converted to a discrete form. For each column, the values are assigned an integer label defining intensity of the measurement compared to the distribution. The quartiles were used as delimiters of 4 different classes of intensity, as shown in Table 2 (1=Very Low, 4=Very High):

Label Id	Range
1	min – 25%
2	25% – 50%
3	50% – 75%
4	75% – max

Table 2: Label intervals based on data quartiles

The advantage of numeric over ordinal labels in this situation is enabling working with cumulated range of values (e.g. >= 3 could mean "moderate or high").

The dataset was then exported as an SQL table with the following schema:

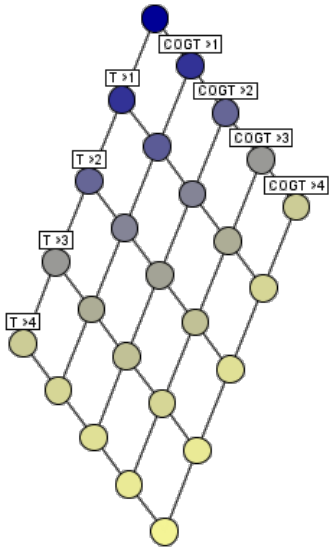
```
CREATE TABLE mytable(
   Timestamp  VARCHAR(19) NOT NULL PRIMARY KEY,
   COGT       INTEGER  NOT NULL,
   NOxGT      INTEGER  NOT NULL,
   T          INTEGER  NOT NULL,
   RH         INTEGER  NOT NULL,
   PT08S1CO   INTEGER  NOT NULL,
   PT08S2NMHC INTEGER  NOT NULL,
   PT08S3NOx  INTEGER  NOT NULL,
   PT08S4NO2  INTEGER  NOT NULL,
   PT08S5O3   INTEGER  NOT NULL
);
```

3.2 Defining scales

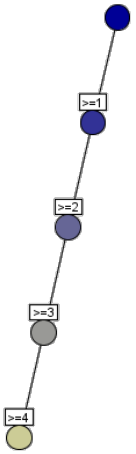
A subset of the columns were used in creating a number of scales, whose concept lattices are visible in Figure 10. Namely, I have defined:

- CO (ground truth) versus Temperature grid scale
- Measurements of a CO sensor in an ordinal scale, including bounds
- Measurements of a NOx sensor in an ordinal scale, including bounds
- Relative humidity in a two-way ordinal scale, where increasing scale includes bounds
- A combined pollution scale as attributes list, based on a heuristic implying multiple ground truths chemical values:
  - Low: COGT ¡= 2 AND NOxGT ¡= 1
  - Moderate: (COGT = 3 OR NOxGT = 2) AND NOT (COGT = 4 OR NOxGT = 4)
  - High (COGT = 4 OR NOxGT = 3) AND NOxGT ¡¿ 4
  - Very High: NOxGT = 4

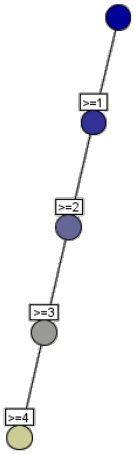
- A seasons scale, extracted from the timestamp (Spring, Summer, Autumn, Winter)



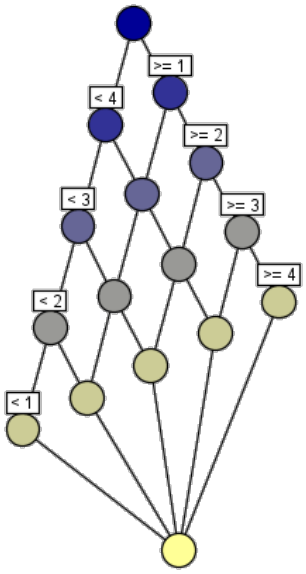
(a) CO vs Temperature



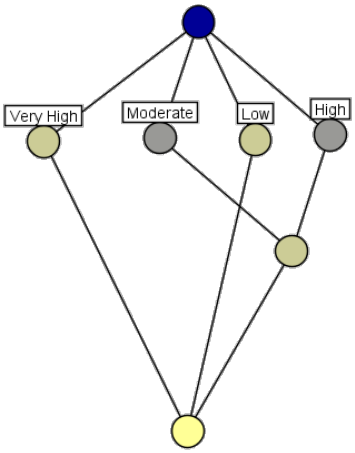
(b) PT08S1_CO



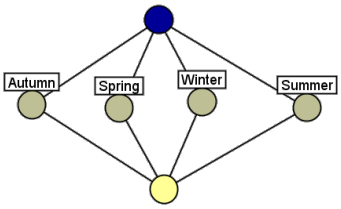
(c) PT08S1_NOx



(d) Relative Humidity



(e) Pollution



(f) Seasons

Figure 10: Context scales

3.3 Exploring with ToscanaJ

Let's take a high level look at pollution level in each season (Figure 11). Some primordial knowledge can be extracted, such as the fact that the colder seasons see higher levels of pollution (they have most Very High and Moderate-High measurements). On the other side, Summer is the season that sees the lowest amount of pollution.

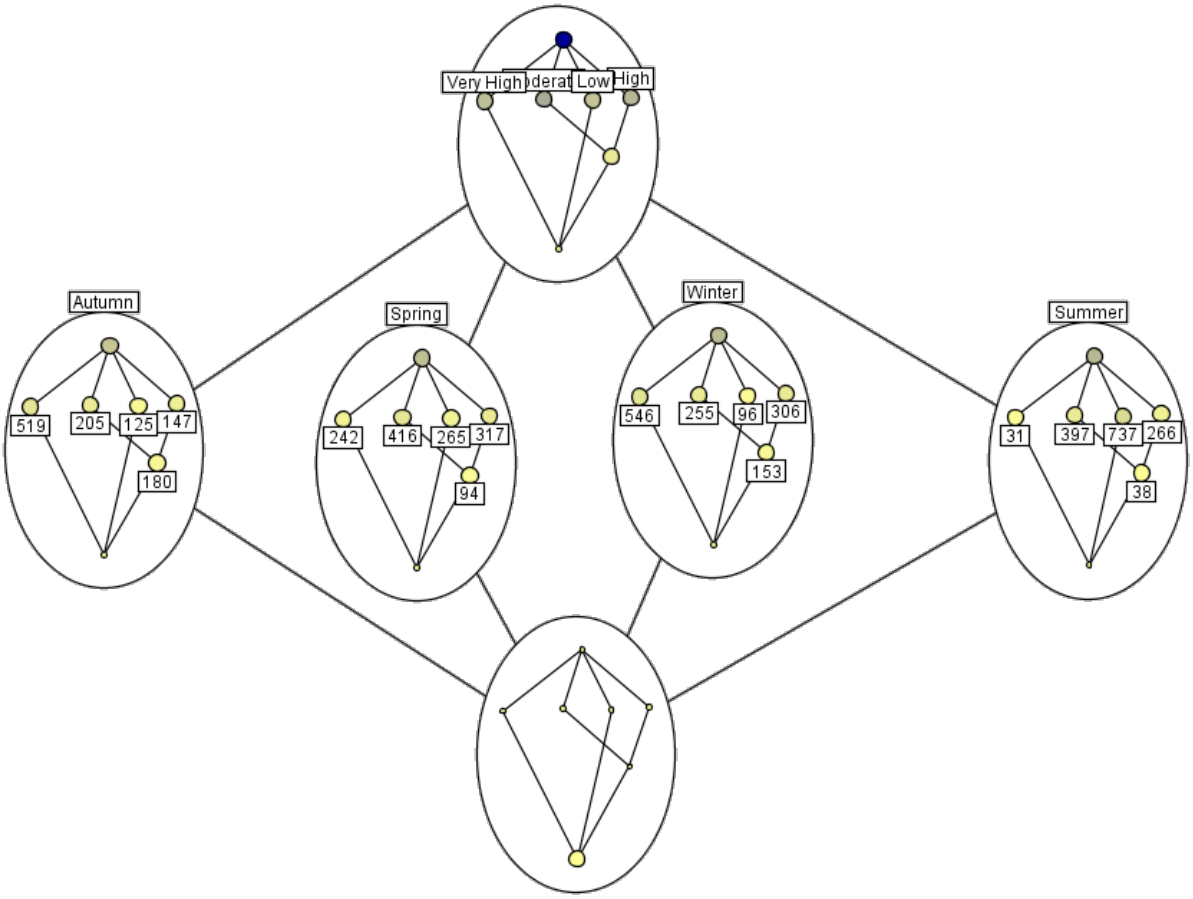


Figure 11: Pollution in each season – nested diagram

Next, we can focus our attention on the relationship of relative humidity (RH) and sensor measurements (Figures 12 and 13). The concept lattices reveal several patterns in the relationship between relative humidity and pollutant sensor readings. When humidity is low, CO values tend to remain low (typically <= 2), while increasing humidity corresponds to a slight rise in CO concentrations, with higher humidity levels showing broader variability, suggesting less predictability or potentially higher values. For NOx, the trend is not strongly pronounced: while low humidity is associated with generally low NOx readings (e.g., <= 2), increasing humidity does not consistently lead to higher NOx concentrations. In fact, under very high humidity, the highest NOx values appear less dominant, suggesting that high humidity may slightly suppress peak NOx levels, possibly due to dispersion or chemical absorption in moist air. In contrast, CO levels show a clearer pattern of gradual increase with rising humidity, although the relationship is not strictly linear and may plateau or vary. Environmentally, the results imply that CO concentrations are more sensitive to humidity than NOx levels, and any potential interaction between humidity and NOx appears to be weak or indirect.

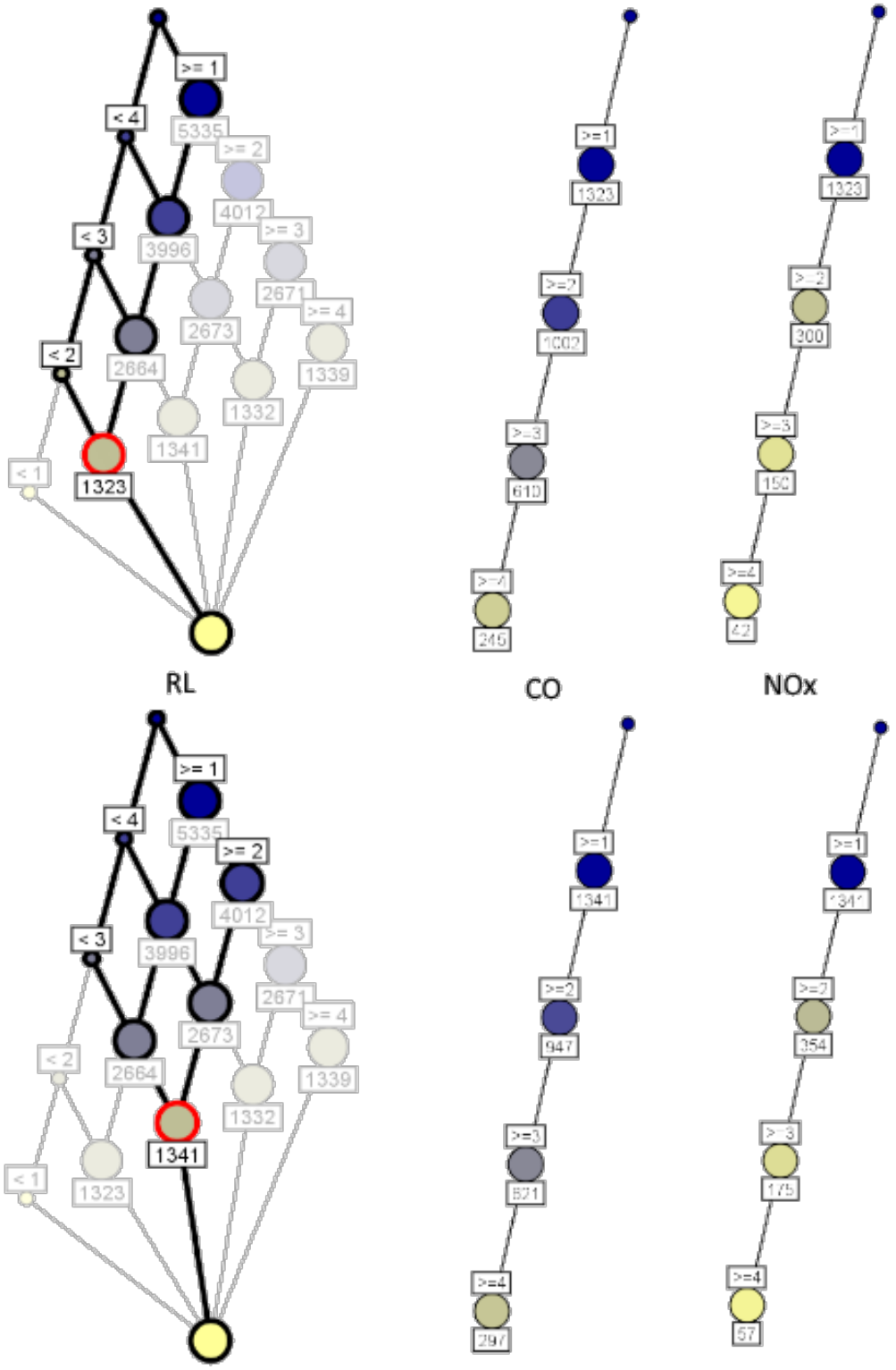


Figure 12: Sensor measurements depending in relative humidity (1)

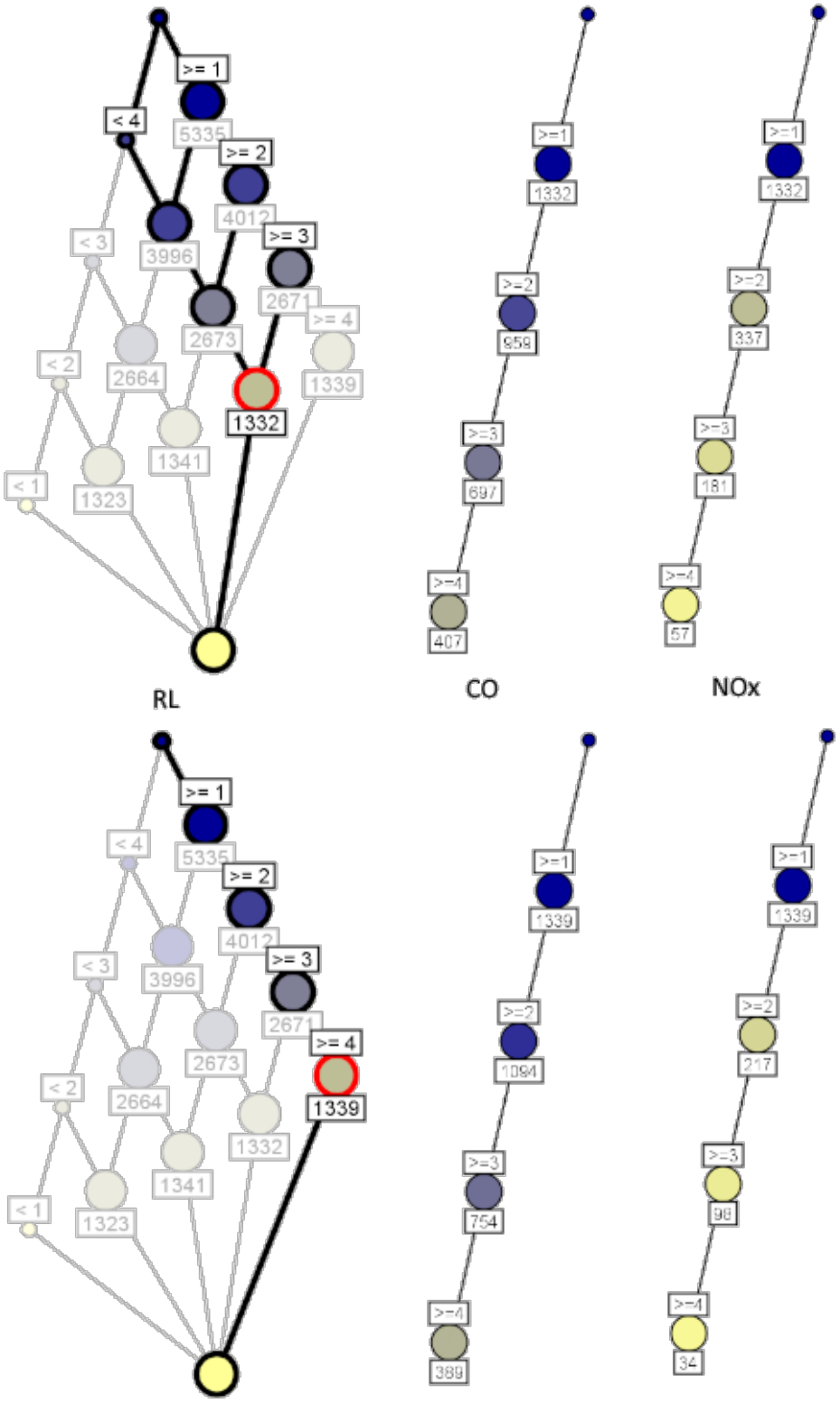


Figure 13: Sensor measurements depending in relative humidity (2)

From a formal concept analysis perspective, the depth and refinement of concept paths in each lattice

illustrate ToscanaJ's strength in navigating increasing attribute specificity, enabling a nuanced understanding of how environmental factors condition pollutant sensor readings.understanding of how environmental factors influence sensor readings.

4 Attribute exploration

The dataset was brought to binary matrix form and imported into conexp. For simplicity, columns PT08.S2(NMHC) and PT08.S5(O3) were dropped as they are not directly linked to the pollutants we focus on (CO and NOx). Furthermore, the discretization granularity was increased from 4 to 2 (the only labels per column are now 0=Low, 1=High, so 2 attributes for each column). Context reduction was employed. The number of found concepts was 1104. The concept lattice has 4748 edges and a height of 8.

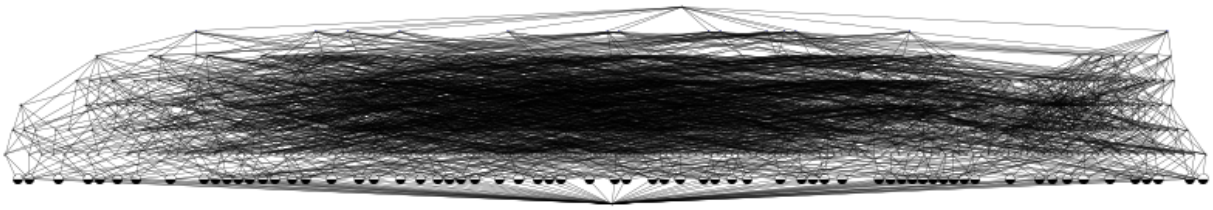


Figure 14: The concept lattice

While performing attribute exploration I realized my limited knowledge about the field which made it impossible to answer the expert questions (Figure 15).

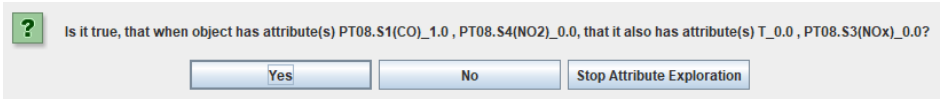


Figure 15: Example of difficult expert question

Therefore, I reverted to an easier context from a more familiar field. The context table can be seen below in table 3.

Language	**Stat**	**GC**	**FP**	**OOP**	**LowLvl**	**Web**	**Safe**
Python	0	1	1	1	0	1	1
Java	1	1	0	1	0	1	1
C	1	0	0	0	1	0	0
JavaScript	0	1	1	0	0	1	1
Haskell	1	1	1	0	0	0	1
Rust	1	0	1	1	1	0	1
C++	1	0	0	1	1	0	0
Go	1	1	0	0	1	1	1
Kotlin	1	1	1	1	0	1	1
TypeScript	1	1	1	1	0	1	1
Swift	1	1	0	1	1	0	1
Ruby	0	1	1	1	0	1	1
PHP	0	1	0	1	0	1	1
C#	1	1	1	1	0	1	1

Table 3: Programming languages context table

The new number of concepts is 34 (Figure 16).

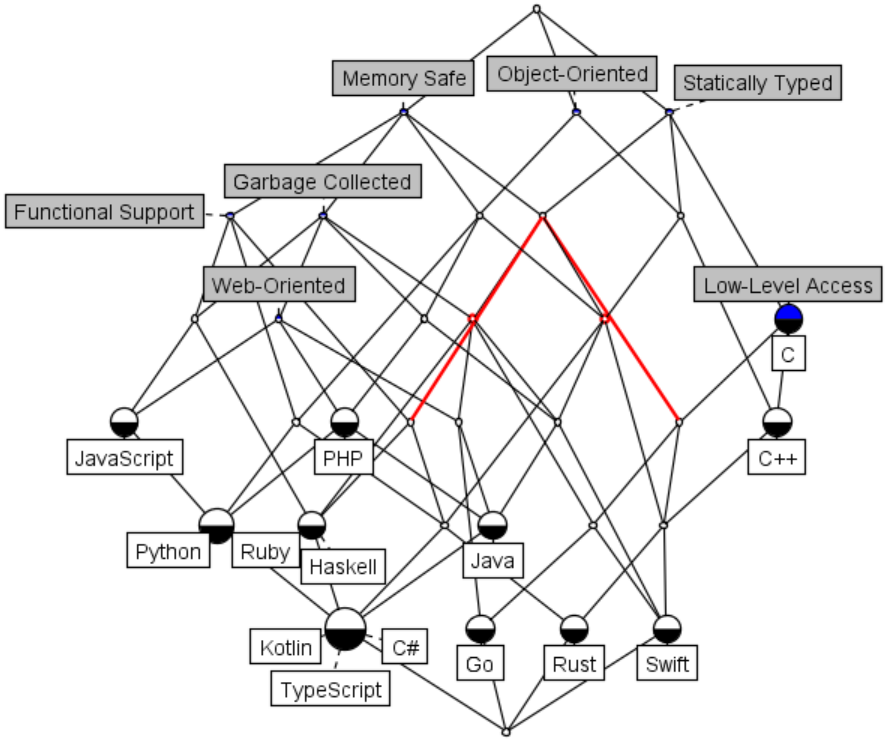


Figure 16: Concept lattice for the programming languages context

Attribute exploration implications:

- Web-Oriented = Garbage-Collected, Memory Safe? (Yes)
- Low-Level Access = Statically Typed? (Yes)
- Functional Support = Memory Safe? (Yes - lack of counter-examples)
- Garbage Collected = Memory Safe? (Yes)
- Garbage Collected, Functional Support, Object Oriented, Memory Safe = Web Oriented? (Yes - most of the time)
- Statically Typed, Functional Support, Low-Level Access, Memory Safe = Object Oriented? (Yes)
- Statically Typed, Garbage Collected, Object Oriented, Low-Level-Access, Web Oriented, Memory Safe = Functional Support? (Yes - potentialy false precondition: conflicting Low-Level Accesn and Web Oriented)
- Statically Typed, Garbage Collected, Functional Support, Web-Oriented, Memory Safe =¿ Object-Oriented? (Yes)
- Statically Typed, Garbage Collected, Object-Oriented, Low-Level Access, Web-Oriented, Memory Safe = Functional Support? (Yes)
- Statically Typed, Garbage Collected, Functional Support, Web Oriented, Memory Safe = Object Oriented? (Yes)

From the attribute exploration, we learn that programming languages designed for web environments are consistently garbage-collected and memory safe, while low-level access strongly correlates with being

statically typed. Functional support generally implies memory safety, and garbage collection almost always ensures it as well. Languages that combine garbage collection, functional programming, object orientation, and memory safety tend to be web-oriented. Additionally, in the observed dataset, statically typed languages that support functional programming, offer low-level access, and ensure memory safety are also object-oriented. However, some implications—such as those involving both low-level access and web orientation—may be less robust due to their conflicting nature. Overall, these relationships highlight how type systems, memory management, programming paradigms, and typical application domains are closely interconnected in real-world languages.

Moreover, by computing the association rules, we can further confirm the strong interdependence between key programming language attributes. High-confidence rules indicate that memory safety is closely linked to garbage collection and object orientation, with over 80–90% confidence that languages featuring memory safety also support garbage collection and object-oriented paradigms. Similarly, functional support paired with memory safety often implies garbage collection and web orientation. Statically typed languages strongly associate with memory safety and object orientation as well. Notably, the presence of combinations like statically typed, garbage collected, object-oriented, and memory safe often predict web orientation and functional support with high confidence. Overall, these rules quantitatively reinforce the attribute correlations observed earlier, highlighting how modern programming languages tend to cluster around particular sets of features reflecting common design goals and application domains.

5 Knowledge discovery in triadic context

For this task I chose MovieLens Latest Small dataset [2]. For 3-FCA, I chose objects to be users, attributes are movies, and conditions are ratings. I selected a subset of 4 users and 4 movies for ease of computation. The ratings were classified in 3 ranges, signifying a low, medium or high rating. The final context table can be seen in figure 17.

x 1	x 296	x 318	x 356	x 593
x 414				
x 448			x	
x 474			x	
x 599			x	x

x 2	x 296	x 318	x 356	x 593
x 414				
x 448				
x 474	x			x
x 599		x		

x 3	x 296	x 318	x 356	x 593
x 414	x	x	x	
x 448	x			x
x 474		x		
x 599	x			

Figure 17: Triadic context table

5.1 Building the concepts

Formally, the triadic context $\mathbf{F} = \{U, M, R, Y\}$, where: $U = \{U_{414}, U_{448}, U_{474}, U_{509}\}$,
$M = \{M_{296}, M_{318}, M_{356}, M_{593}\}$,
$R = \{R_1, R_2, R_3\}$,
$Y = \{(U_{448}, M_{356}, R_2), (U_{474}, M_{356}, R_2), (U_{599}, M_{593}, R_2), (U_{599}, M_{356}, R_2), (U_{414}, M_{593}, R_3)$,
$(U_{474}, M_{593}, R_3), (U_{474}, M_{296}, R_3), (U_{599}, M_{318}, R_3), (U_{414}, M_{356}, R_4), (U_{414}, M_{318}, R_4), (U_{414}, M_{296}, R_4)$,
$(U_{448}, M_{593}, R_4), (U_{448}, M_{296}, R_4), (U_{474}, M_{318}, R_4), (U_{599}, M_{296}, R_4)\}$

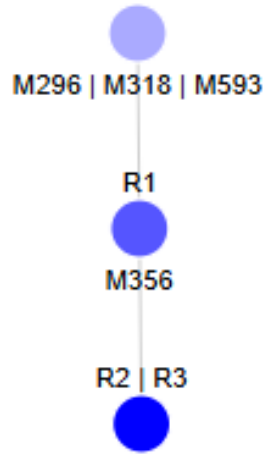
11 concepts are found:

- 1. $\{U_{414}, U_{448}, U_{474}, U_{599}\}$, $\{\}$, $\{R_1, R_2, R_3\}$
- 2. $\{U_{599}\}, \{M_{318}\}, \{R_2\}$
- 3. $\{U_{474}\}, \{M_{296}, M_{593}\}, \{R_2\}$
- 4. $\{\}, \{M_{296}, M_{318}, M_{356}, M_{593}\}, \{R_1, R_2, R_3\}$
- 5. $\{U_{448}, U_{474}, U_{599}\}, \{M_{356}\}, \{R_1\}$

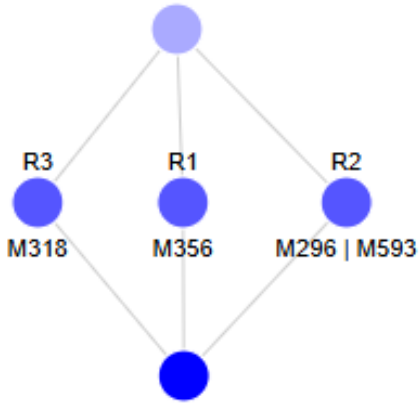
[2] https://www.kaggle.com/datasets/grouplens/movielens-latest-small

- 6. $\{U_{599}\}, \{M_{356}, M_{593}\}, \{R_1\}$
- 7. $\{U_{414}, U_{448}, U_{599}\}, \{M_{296}\}, \{R_3\}$
- 8. $\{U_{414}, U_{474}\}, \{M_{318}\}, \{R_3\}$
- 9. $\{U_{414}\}, \{M_{296}, M_{318}, M_{256}\}, \{R_3\}$
- 10. $\{U_{448}\}, \{M_{296}, M_{593}\}, \{R_3\}$
- 11. $\{U_{414}, U_{448}, U_{474}, U_{599}\}, \{M_{296}, M_{318}, M_{356}, M_{593}\}$

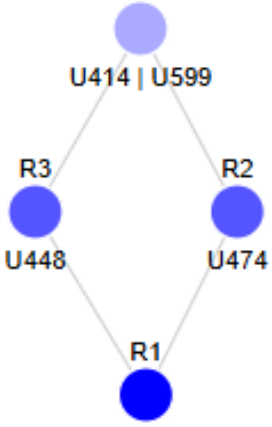
5.2 Local navigation



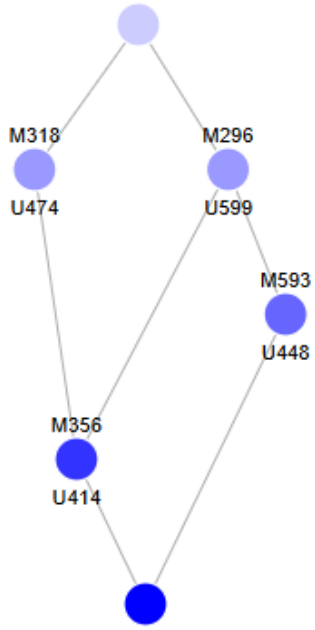
(a) Lock $\{U_{448}, U_{474}, U_{599}\}$



(b) Lock $\{U_{474}\}$



(c) Lock $\{M_{296}, M_{593}\}$



(d) Lock $\{R_3\}$

Figure 18: Triadic concept lattice projections

Some lock lattices are showcased in Figure **??**. In subfigure (a), the projection fixes a subset of users and visualizes how they relate to media and ratings. It shows that these users commonly interact with media like M356, M296, M318, and M593 under various ratings (R1–R3). The structure highlights shared access patterns, particularly emphasizing M356 as a common medium linking multiple ratings. The view in subfigure (b) isolates a single user, U474, and maps out their rating-based interactions with media. It reveals that U474 rated M296 and M593 with R2, M318 with R3, and M356 with R1. The projection illustrates

how one user engages with different media across varying ratings, reflecting a diverse evaluation profile. In subfigure (c), by fixing these two media items, the projection explores which users rated them and with what ratings. It shows that U414, U448, U474, and U599 all interacted with M296 and/or M593 across ratings R1–R3. The lattice highlights common user evaluation patterns for these media, suggesting popular or central content. The projection in subfigure (d) fixes rating R3 and examines user-media interactions within this rating. It demonstrates that users like U414, U448, U474, and U599 rated media such as M296, M318, and M593 with R3. The hierarchy reflects which users share content preferences under this specific rating level.

6 Temporal Concept Analysis

We follow Frodo at five major points in his quest, focusing on his mental and physical state, companionship, and progress toward destroying the One Ring.

The time granules are $G = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$, the labels T_i are defined in table 4 and correspond to a moment of a specific day.

Label	**Day**	**Time of Day**	**Story Moment**
T1	Day 1	Morning	Frodo sets out from the Shire with companions and hope.
T2	Day 2	Evening	The group reaches the forest edge; tension rises.
T3	Day 4	Morning	At a quiet camp, Frodo feels the Ring's growing weight.
T4	Day 5	Evening	Into dark woods; danger and doubt deepen.
T5	Day 6	Morning	Frodo and Sam enter enemy lands, alone and burdened.
T6	Day 7	Afternoon	The journey grows harsher; hope fades.
T7	Day 10	Morning	At Mount Doom's base; despair sets in.
T8	Day 10	Evening	On the brink of collapse in the shadow of the volcano.
T9	Day 12	Morning	Frodo stands alone inside Mount Doom.
T10	Day 14	Evening	The Ring is destroyed; Frodo collapses.

Table 4: Timeline of key story moments

The event attributes and their values are defined in table 5.

Attribute	**Possible Values**
ring_burden	light / growing / heavy
companions	many / few / only Sam
danger_level	low / medium / high
hope	strong / wavering / faint
location_type	home / haven / peril / enemy_land / doom_site

Table 5: Attributes and their possible values

The many values context is showcased in table 6.
The lattices for the time part (T) and event part (C) can be consulted in figures 19 and 20.

Timestamp	Day	Time of Day	ring_burden	companions	danger_level	hope	location_type
T1	Day 1	Morning	light	many	low	strong	home
T2	Day 2	Evening	light	many	medium	strong	forest_edge
T3	Day 4	Morning	growing	few	medium	wavering	safe_camp
T4	Day 5	Evening	growing	few	high	wavering	dark_forest
T5	Day 6	Morning	heavy	only Sam	high	faint	enemy_territory
T6	Day 7	Afternoon	heavy	only Sam	high	faint	enemy_territory
T7	Day 10	Morning	heavy	only Sam	critical	despair	doom_site
T8	Day 10	Evening	heavy	only Sam	critical	despair	doom_site
T9	Day 12	Morning	heavy	none	critical	despair	doom_site
T10	Day 14	Evening	heavy	none	critical	despair	doom_site

Table 6: Timeline and event attribute values

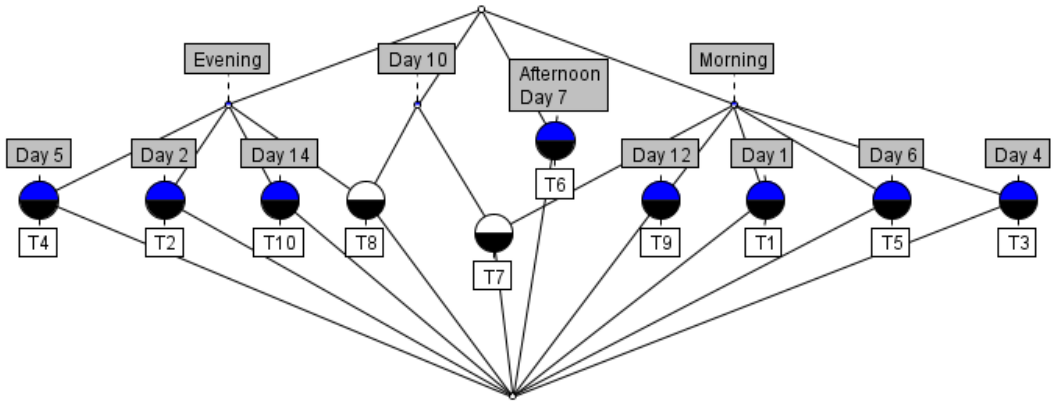


Figure 19: Lattice of time part

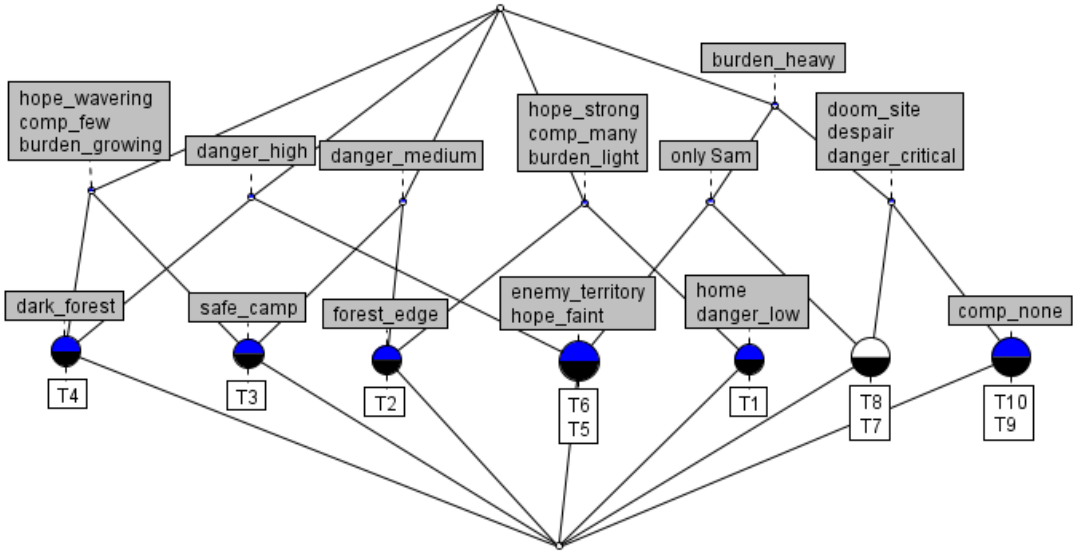


Figure 20: Lattice of event part

The derived context shown in Figure 21 enables the construction of a comprehensive situation space. This space reveals the intersections between time-related and event-related attributes. Each node in this lattice represents a conceptual situation involving both when (time granule) and what (event conditions). For example, timestamps T5 and T6 share a concept node that reflects a sustained state of high danger and faint hope with only one remaining companion.

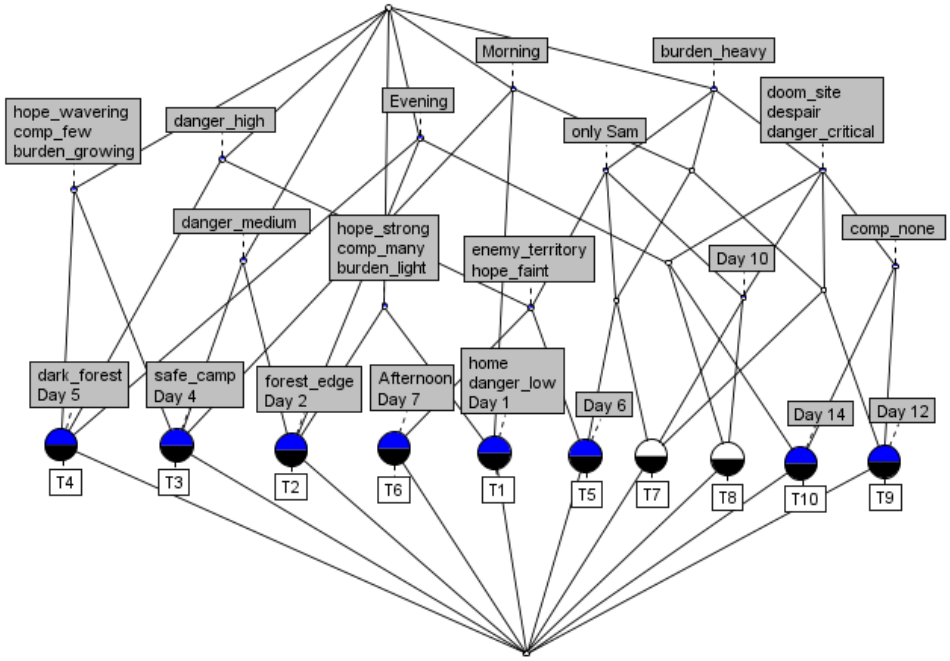


Figure 21: Lattice of the derived context

The time relation is $R = \{(T_i, T_{i+1}) | i = \overline{1,9}\}$ and the transitions are graphically represented in figure 22. These transitions reflect the natural ordering of time in our dataset: a directed graph where each node (T1 to T10) is a timestamp and edges indicate succession. This ordering underlies all subsequent transition and lifetrack analysis.

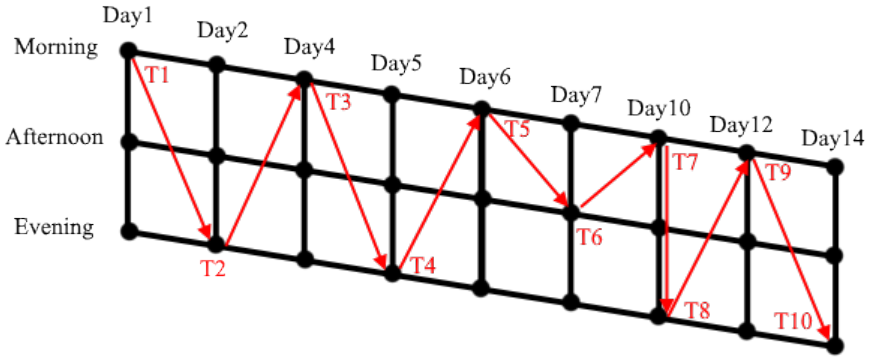


Figure 22: Time schedule

The Lifetrack of companion states (figure 23 maps the sequence of changing states with respect to companions (from “many” to “none”) over time. This visualization illustrates the temporal degradation of social support in the journey. Initially, the individual is supported by many companions, but over time, this

support dwindles to only Sam and then to none. Each red arrow denotes a transition in the “companions” attribute within the conceptual time system.

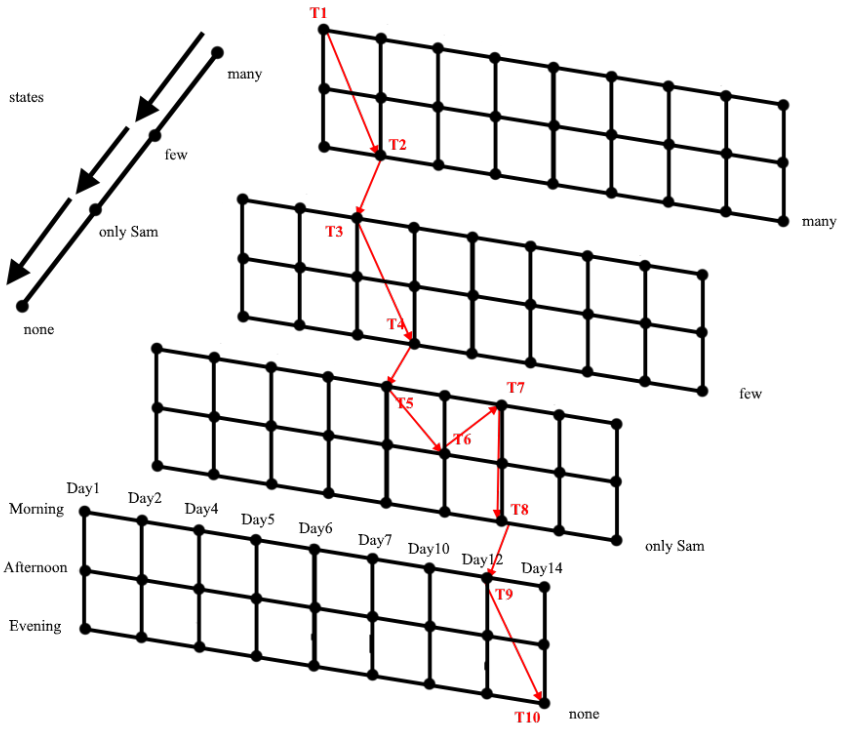


Figure 23: Lifetrack of companions states

The life track represented in Figures 22 and 23 visualizes the conceptual trajectory of the subject through time and space. The journey begins from a state of hopefulness, low danger, and strong social support (timestamp T1), and progresses through a series of increasingly difficult and deteriorating conditions. By the midpoints (such as T3 to T6), we observe rising burden, growing danger, and weakening emotional resilience, culminating in states of isolation, critical danger, and despair by the end (T7 to T10). This evolution represents a monotonic decline, both emotionally and physically, which is effectively captured by the concept lattice structure and lifetrack visualization.

Temporal Concept Analysis reveals that states are temporally grounded conceptual objects, and each time granule is uniquely mapped to one of these states. The event part lattice provides semantic categories for these states, such as ”safe”, ”dangerous”, or ”critical”, while the time part lattice encodes structural granularity of time. The latter shows how situations are grouped under higher-level temporal categories like ”Morning” or ”Day 10”.

This application of Temporal Concept Analysis (TCA) demonstrates how the technique enables a structured understanding of dynamic, temporal processes. Through a combination of event and time lattices, derived context modeling, and life track visualization, we gain insight into the subject’s temporal evolution through complex and multidimensional states.

The analysis reveals consistent patterns, such as the deterioration of physical safety, emotional stability, and social support. By combining the formal apparatus of Formal Concept Analysis with temporal ordering and granularity, TCA provides a powerful framework for visualizing, interpreting, and reasoning about the progression of conceptual states over time. This example illustrates the utility of TCA in domains requiring a fine-grained understanding of temporal evolution, whether narrative, psychological, or process-driven in nature.

7 A Review of Applications of Formal Concept Analysis in Computer Vision

7.1 Introduction

Formal Concept Analysis (FCA) is a mathematical theory for data analysis that provides a principled way to derive conceptual hierarchies from data. It was developed by Rudolf Wille in the 1980s, FCA has its roots in lattice theory and aims to represent knowledge in a structured manner by identifying "formal concepts" within a given dataset [4]. A formal concept is defined as a pair of an extent (a set of objects) and an intent (a set of attributes) such that the extent consists of all objects that possess all attributes in the intent, and the intent consists of all attributes shared by all objects in the extent. This duality allows for constructing a concept lattice, which visually represents the hierarchical relationships between these concepts.

The importance of FCA lies in its ability to provide a clear, interpretable, and mathematically sound framework for knowledge representation, data mining, and information retrieval. It has applications in diverse fields, such as software engineering, chemistry, biology, medicine, and information science, where it is used for tasks like data classification, clustering, knowledge discovery, and ontology building [4]. The inherent structure of concept lattices makes FCA particularly valuable for exploring complex datasets and uncovering hidden relationships. A typical concept lattice visually organizes concepts from general to specific, as conceptually illustrated in Figure 24.

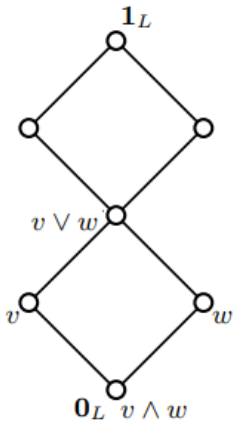


Figure 24: Conceptual illustration of a Formal Concept Lattice. [4]

7.2 Related Work

In recent years, the principles of FCA have found increasing relevance in the field of computer vision. Computer vision deals with the challenging task of enabling computers to "see" and interpret visual information from the real world. This often involves processing vast amounts of unstructured image and video data, considerably increasing the need for robust knowledge representation and retrieval mechanisms. FCA provides a unique way to organize, browse, and retrieve visual content by establishing conceptual links between images based on their shared features or annotations. For example, FCA has been explored for information extraction from document images, where it can assist in template detection and knowledge graph rule induction, proving its utility in structuring visual data for specific tasks [5]. This capability to derive meaningful structures from visual data makes FCA a promising tool for addressing various challenges in computer vision, from image retrieval to scene understanding.

7.3 Review of ImageSleuth

ImageSleuth is a notable application that uses Formal Concept Analysis for organizing, browsing, and searching image collections. It represents a significant effort to apply the rigorous mathematical framework of FCA

to the practical challenges of managing large visual datasets. The core idea behind ImageSleuth and similar systems is to transform image metadata or features into a formal context, allowing FCA to generate a concept lattice that can then be used for intuitive navigation and retrieval.

One of the early instantiations of this approach is observed in systems like Camelis, which aimed to organize and browse personal photo collections using a FCA-based logical information system [3]. Camelis demonstrated how FCA could be used to automatically generate a conceptual hierarchy from user-provided annotations or extracted image features. Users could then navigate this hierarchy, moving from more general concepts (e.g., "outdoor photos") to more specific ones (e.g., "photos with trees and people") by traversing the concept lattice. This provided a more flexible and exploratory browsing experience compared to traditional keyword-based search or folder structures. The system allowed users to refine their queries by selecting attributes, and the concept lattice would dynamically update to show relevant images and associated concepts. An example of how such a system presents concepts and allows navigation is conceptually depicted in Figure 25.

Figure 25: Conceptual illustration of browsing a photo collection using a concept lattice with Camelis [3]

ImageSleuth further extended these capabilities, particularly in the context of browsing and searching MPEG-7 images [2]. MPEG-7 is an ISO standard for describing multimedia content, providing a rich set of descriptors for various aspects of images and videos, such as color, texture, shape, and semantic annotations. By mapping these MPEG-7 descriptors to attributes in a formal context, ImageSleuth could construct concept lattices that represent the conceptual relationships within large collections of multimedia content. This allowed users to perform sophisticated queries that combine visual features with semantic annotations, leading to more precise and context-aware retrieval. For example, a user could search for images with a specific color distribution and also containing "sky" or "water" as semantic concepts. The FCA framework naturally supports such multi-faceted queries by identifying concepts that encompass both visual and semantic properties. The process of mapping MPEG-7 descriptors to a formal context is conceptually shown in Figure 26.

```
<VisualDescriptor xsi:type="ColorLayoutType">
   <YDCCoeff>5</YDCCoeff>
   <CbDCCoeff>30</CbDCCoeff>
   <CrDCCoeff>31</CrDCCoeff>
   <YACCoeff63>
      13 23 15 12  5 20  9 14 19 17 16 17 21 18 15 17 18 12 16 11 13
      16 14 15 15 15 17 13 16 15 17 14 20 15 17 16 18 15 16 15 15 12
      14 15 16 15 16 14 16 15 16 16 17 16 15 15 14 15 15 15 16 17 16
   </YACCoeff63>
   <CbACCoeff63>
      16 15 16 16 17 15 16 16 15 15 16 15 15 15 16 15 15 16 15 16 16
      16 16 16 16 16 15 16 16 16 15 16 15 16 15 16 15 16 16 16 16 16
      16 16 15 16 16 16 16 16 15 16 15 15 16 16 16 16 16 16 15 15
   </CbACCoeff63>
   <CrACCoeff63>
      16 16 16 16 16 15 15 16 16 15 16 16 16 15 15 16 16 16 16 15 15
      16 15 16 15 15 15 16 16 16 16 16 15 16 16 16 15 16 16 16 16 16
      15 15 16 16 15 16 15 16 16 16 16 15 15 16 16 16 16 16 16 16
   </CrACCoeff63>
</VisualDescriptor>
```

(a) Mapping of MPEG-7 color descriptors to a formal context for FCA in ImageSleuth [2]

(b) Edge types used in edge histogram descriptor [2]

	Price	Needs::Hunger	Needs::Comfort	Needs::Hygiene	Needs::Bladder	Needs::Energy	Needs::Fun	Needs::Environment	Skills	Function	Room Type
4 by 4 Designer Chandelier	§120	0	0	0	0	0	0	1		Lighting	Dining, Living, Bathroom, Bedroom
Absolutely Nothing Special	§850	0	0	0	0	0	0	1		Lighting	Kids, Study, Dining, Living, Bedroom
Ad-a-Quaint Barstool	§285	0	3	0	0	0	0	1		Comfort	Living, Kitchen
Ad-a-Quaint Coffee Table	§140	0	0	0	0	0	0	1		Surfaces	Study, Living
Astrowonder Telescope	§550	0	0	0	0	0	4	0	Logic	Hobbies	Outside
Zenu Meditation Sleeper	§950	0	4	0	0	4	0	2		Comfort	Bedroom

(c) Samples from item properties context [2]

Figure 26: Visual representations and contexts extracted from MPEG-7 descriptors [2]

The integration of ImageSleuth into real-world applications is exemplified by projects like DVDSleuth [1]. DVDSleuth was presented as a case study in applied Formal Concept Analysis for navigating web catalogs, specifically focusing on DVD collections. While not strictly a computer vision application in the sense of processing raw pixels, it demonstrated how the principles of ImageSleuth could be adapted to manage and explore large item catalogs where items (like DVDs) have numerous attributes (genre, actors, director, year, etc.). By applying FCA to these attributes, DVDSleuth created a concept lattice that could be navigated and allowed users to browse the catalog conceptually, discovering relationships between DVDs that might not be apparent through simple keyword searches. This showcased the versatility of the FCA-based approach, proving its utility beyond pure image content to any domain where structured browsing and conceptual navigation of attributed objects are beneficial. The success of these systems highlights FCA's strength in providing a robust and intuitive framework for information organization and retrieval, particularly when dealing with complex, multi-attributed data like image collections.

The integration of ImageSleuth into real-world applications is exemplified by projects like DVDSleuth [1]. DVDSleuth was presented as a case study in applied Formal Concept Analysis for navigating web catalogs, specifically focusing on DVD collections. While not strictly a computer vision application in the sense of processing raw pixels, it demonstrated how the principles of ImageSleuth could be adapted to manage and explore large item catalogs where items (like DVDs) have numerous attributes (genre, actors, director, year, etc.). By applying FCA to these attributes, DVDSleuth created a concept lattice that could be navigated and allowed users to browse the catalog conceptually, discovering relationships between DVDs that might not be apparent through simple keyword searches. This showcased the versatility of the FCA-based approach, proving its utility beyond pure image content to any domain where structured browsing and conceptual navigation of attributed objects are beneficial. The success of these systems highlights FCA's strength in providing a robust and intuitive framework for information organization and retrieval, particularly when dealing with complex, multi-attributed data like image collections.

7.4 Discussions

The evaluation of FCA-based approaches in computer vision, particularly systems like ImageSleuth and its predecessors, often focuses on their effectiveness in facilitating information retrieval, browsing, and knowledge discovery, rather than traditional machine learning metrics like classification accuracy. The primary goal is to enhance user experience and efficiency in navigating large, complex datasets.

In the context of Camelis, which organized personal photo collections, the evaluation highlighted the system's ability to provide a flexible and intuitive browsing experience [3]. Camelis was successfully applied to a personal photo collection containing more than 5,000 photos [3]. The system is explicitly stated to be efficient for collections comprising up to 10,000 objects [3]. A critical aspect of Camelis's efficiency lies in the computation of its navigation trees, which operates in linear time relative to the size of the context ($O(N)$) [3]. This linear complexity for navigation tree computation ensures responsiveness and interactivity, directly supporting the system's user-centric design that encourages metadata input by making the effort immediately worthwhile [3].

For ImageSleuth's application to MPEG-7 images, the evaluation centered on its capability to effectively browse and search multimedia content based on rich descriptors [2]. The evaluation of the IMAGE-SLEUTH program was conducted through usability testing involving 29 subjects [2]. The test collection utilized for this study comprised a subset of MPEG-7 images created from the computer game The Sims 2™ [2]. While the precise number of images within this subset is not explicitly quantified in the provided documentation, the context implies a collection size suitable for a comprehensive usability trial. An experiment with the 29 testers provided specific comparative performance results for local navigation on concept lattices (Galois Lattices or GLs): local navigation on GLs outperforms hierarchical classification navigation, but it does not perform better when directly compared to Boolean querying [2]. This outcome indicates that while concept lattice navigation offers distinct advantages over rigid hierarchical structures for exploratory tasks, it does not surpass the efficiency of direct Boolean queries for specific, well-defined search objectives. A significant limitation identified for concept lattice-based systems, including ImageSleuth, pertains to the scalability of visualizing the entire concept lattice. Navigation based on concept lattices has traditionally been used with relatively small datasets, typically involving only a few dozen to a few hundred objects [2]. In addition, rendering the full structure of a concept lattice becomes increasingly impractical when the number of objects exceeds several dozen [2]. The number of generated concepts can grow rapidly; for example, a dataset containing around one thousand objects may theoretically produce up to one million distinct concepts [2]. To address these challenges, a widely adopted and efficient strategy is to show only the immediate parent and child concepts relative to the current concept, allowing for the calculation and display of relevant local neighborhoods in a time-efficient manner [2]. The system also restricts navigation by disallowing movement to concepts that contain no associated images or to the lowest-level concept if it lacks any content, which is ensured by preventing users from simultaneously selecting attributes that are logically incompatible [2].

The case study of DVDSleuth, while not directly a computer vision application, provides insights into the practical applicability and evaluation of FCA-based systems for navigating large catalogs [1]. DVDSleuth applied the same conceptual representation, navigation, and clustering techniques as ImageSleuth to an information space built from a dynamic collection sourced from the Amazon.com online store [1]. A concrete number for the size of the specific dataset or subset of DVDs used in the case study's evaluation is not explicitly provided in the snippets. The application of FCA techniques to a dynamic collection sourced from Amazon.com implicitly validates the effectiveness of local navigation and other partial lattice visualization strategies for large-scale contexts.

Overall, the evaluation of these FCA-based systems in computer vision and related domains emphasizes the qualitative benefits of conceptual navigation, the flexibility of query formulation, and the ability to discover hidden relationships within data. While direct quantitative comparisons with other retrieval methods (e.g., precision/recall curves) are less prominent in the provided literature, the consistent theme is that FCA provides a powerful and intuitive framework for organizing and accessing complex information, leading to enhanced user interaction and knowledge discovery. The numbers often refer to the size of the datasets handled or the complexity of the conceptual structures generated, rather than direct performance metrics in a competitive benchmark. A summary of the discussed systems and their key characteristics is provided in Table 7.

Table 7: Summary of FCA-based Systems for Information Organization and Retrieval

System	Primary Purpose	Data Type / Input	Key FCA Application
Camelis [3]	Personal photo organization	User annotations, image features	Generating conceptual hierarchies for browsing
ImageSleuth [2]	Browsing/searching MPEG-7 images	MPEG-7 descriptors (visual/semantic)	Constructing concept lattices for multimedia retrieval
DVDSleuth [1]	Navigating web catalogs (DVDs)	Item attributes (genre, actors, etc.)	Conceptual browsing and discovery in product catalogs

7.5 Conclusion

Formal Concept Analysis offers a robust and mathematically grounded framework for structuring and navigating complex datasets, and its applications in computer vision demonstrate its significant potential. As highlighted by systems like Camelis and ImageSleuth, FCA provides an intuitive method for organizing image collections, facilitating both browsing and searching through the generation of concept lattices. These lattices allow users to explore conceptual relationships between images based on shared attributes, whether derived from manual annotations, extracted visual features (like MPEG-7 descriptors), or metadata.

The strength of FCA in computer vision lies in its ability to transform unstructured visual data into a meaningful, hierarchical knowledge representation. This enables more flexible and powerful query formulation, moving beyond simple keyword matching to conceptual navigation. While quantitative performance metrics like retrieval accuracy are not always the primary focus of evaluation, the qualitative benefits of enhanced user experience, improved browsing efficiency, and the discovery of implicit relationships within image collections are consistently emphasized. The case studies, including the adaptation of ImageSleuth principles for web catalogs like DVDSleuth, underscore the versatility and practical applicability of FCA across various domains requiring structured information access.

In conclusion, FCA provides a valuable paradigm for addressing challenges in computer vision related to content organization, retrieval, and knowledge discovery. Its ability to derive interpretable conceptual structures from diverse data sources makes it a powerful tool for building intelligent systems that can effectively manage and interact with large volumes of visual information. information.

References

[1] Jon Ducrou. Dvdsleuth: A case study in applied formal concept analysis for navigating web catalogs. In *Conceptual Structures: Knowledge Architectures for Smart Applications: 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, July 22-27, 2007. Proceedings 15*, pages 496–500. Springer, 2007.

[2] Jon Ducrou and Peter W Eklund. Browsing and searching mpeg-7 images using formal concept analysis. In *Artificial Intelligence and Applications*, pages 317–322, 2006.

[3] Séastien Ferré. Camelis: Organizing and browsing a personal photo collection with a logical information system. In *Int. Conf. Concept Lattices and Their Applications*, volume 331, pages 112–123, 2007.

[4] Dmitry I Ignatov. Introduction to formal concept analysis and its applications in information retrieval and related fields. In *Russian Summer School in Information Retrieval*, pages 42–141. Springer, 2014.

[5] Mouli Rastogi, Syed Afshan Ali, Mrinal Rawat, Lovekesh Vig, Puneet Agarwal, Gautam Shroff, and Ashwin Srinivasan. Information extraction from document images via fca-based template detection and knowledge graph rule induction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 558–559, 2020.