Utrecht
University

Faculty of Natural Sciences
Master of Science in Artificial Intelligence

Master Thesis

Could you please ClAIrify?: A clustering based framework for machine learning model evaluation

Presented by:
Oscar Alexander Kirschstein Schafer

Academic course:
2022-2023

Could you please ClAIrify?: A clustering based framework for machine learning model evaluation

Oscar Alexander Kirschstein Schafer

Oscar Alexander Kirschstein Schafer
Could you please ClAIrify?: A clustering based framework for machine learning model evaluation.

Master of Science in Artificial Intelligence.
Faculty of Natural Sciences.
University of Utrecht.

Master Thesis. Academic course 2022-2023.

Advisor(s)

Yuncong Yu
ViG Research Group

Dr. Michael Behrisch
ViG Research Group

Dr. Alex Telea
ViG Research Group

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence

ARIMA Autoregressive Integrated Moving Average

DBSCAN Density-Based Clustering Algorithm

DTW Dynamic Time Warping

GDP Gross Domestic Product

GUI Graphic User Interface

JSON Javascript Object Notation

LR Linear Regression

LSTM Long-Short Term Memory

MAPE Mean Average Performance Error

ML Machine Learning

PCA Principal Component Analysis

RMSE Root Mean Square Error

RQ Research Question

SRQ Secondary Research Question

t-SNE t-distributed Stochastic Neighbor Embedding

TAP Think Aloud Protocol

UMAP Uniform Manifold Approximation and Projection

UMUX Usability Metric for User Experience

VA visual analytics

VAEI Per-Value Added Energy Intensity

Abstract

Machine Learning and Visual Analytics have received increasing attention in recent years both in research and production. These two fields overlap on the evaluation of model performance. Classical evaluation metrics have limited explainability, and do not use the visualisation potential available nowadays, nor are they able to give insights on how models perform on different types of sub-groups of the input data. Our goal for this project is to present a data-, task-, and model-agnostic framework to overcome these limitations of classical evaluation metrics, capable of generating visual model performance profiles over clusters of the evaluation data of a machine learning problem. After laying out the building blocks of this conceptual framework in the form of guidelines, we implement it as a web app for demonstration and validation. Finally, we go on to validate the quality of the framework through a Think Aloud Protocol study, a survey and two use cases on time series data sets. Thematic analysis is then performed on the transcripts resulting from the study. The results point towards the viability of using our conceptual framework for understanding, evaluation and comparison of machine learning model performance.

1 Introduction

In recent years, the convergence of visual analytics (VA) and Machine Learning (ML) has received a lot of attention, both in research and production [18, 37, 67]. VA has been used to improve data analytics with interactive visualisations, a fundamental part of the ML pipeline in every step, from data processing and exploratory data analysis to model selection and comparison. With the help of interactive visual interfaces, users are able to integrate their domain knowledge into interpretation and diagnosis of ML models [20, 45, 58]. The insights that can be obtained through this analysis are important for the understanding and improvement of otherwise oftentimes opaque ML components [33, 37]. Inversely, VA systems can profit from using certain ML algorithms in their workflow to transform big quantities of information into more digestible and explorable visualisations, e.g., through dimensionality reduction or clustering, the methods chosen for this project. In this light, VA and ML mutually benefit each other, and we take advantage of this synergy to tackle a common issue in ML.

1.1 Motivation

A noteworthy step in the ML pipeline is model evaluation, the point in the workflow where it has to be tested if the tackled problem is well solved with the current approach. Not only for validating the approach is this an important step, but also because it is most likely not the last step of the pipeline [50], potentially ushering in a new series of development iterations. To this end, ML practitioners have had, for a long time, a wide selection of metrics at their disposal. [26] The problem with these classical metrics (as we will refer to them), is that they only provide a global view on model performance. They allow for limited interpretability and understanding, while looking at a single model, or while comparing model architectures and/or hyper-parameter configurations. [7, 13, 25, 29]

There are a huge number of different types of models that have been developed recently [39, 41, 55], thanks to a growing research field and computational capacity, but also a huge growth in data to be processed. [40] At the same time, a ML engineer must account for intrinsic variables such as problem constraints, while injecting their domain knowledge. Such a compendium of considerations can quickly make the task daunting.

This is enhanced by the fact that classical evaluation metrics have very limited explainability and do not make use of the visualisation potential available nowadays. There are a multitude of papers that, although with different overarching objectives, point out the shortcomings of classical performance metrics of ML models [8, 12, 24, 38].

Many of these metrics share the characteristic that they do not show the performance trade-offs between different subgroups of the domain data. Although some others, in classification tasks, do try to balance the performance differences in different classes [24, 38], this balancing doesn't say much about the nature of the instances in each class. We conclude that this leads to them having a lower explainability capability, when trying to apply domain knowledge to model development. [12] divides evaluation metrics in two categories: quantitative and qualitative. Quantitative evaluation entails having a way to mechanically measure performance results, while qualitative evaluation has to do with subjective user experiences. The visual medium is the most appropriate for information representation as, if well executed, it can condense valuable information into quickly understandable knowledge [4], and this is why we leverage VA in our approach to bridge the gap between qualitative and quantitative evaluation.

In this light, we propose **ClAIrify**, a data-type-, task- and model-agnostic conceptual framework for model evaluation on different subgroups of clustered input data.

1.2 Goals

The design of the ClAIrify framework is built on two basic principles:

- **Generality**: The framework should have high degrees of liberty for it to be implemented on a wide range of ML problems. Thus we designed it as a conceptual framework consisting of a series of guidelines to be followed in order to create an implementation that is applicable to a specific problem or range of problems.
- **Characterisation**: We aim to make model performance efficiently interpretable, and thus, we have designed ClAIrify's guidelines to work towards this. We want to give the users insight into possible causal relationships between input and output of ML models, considering their performance.

In line with our *generality* principle, we have designed ClAIrify around three agnostic qualities. As such, our framework is: model-agnostic, task-agnostic and data-type-agnostic.

On the other hand, the main way of instantiating the principle of *characterisation* is clustering. We use clustering to divide the evaluation data instances into different homogeneous groups, and provide a view to enable giving semantic meaning to each cluster. In fact, it is this focus on clustering to enable explainability that distinguishes our work from other state-of-the-art solutions in this area.

Clustering serves to find potential inherently homogeneous groupings of data that may not be immediately obvious. This can give insights into how different observations are related to each other. Furthermore, clustering can be paired with data summarising, creating a representative prototype for each cluster, which can give an understandable overview of each of the data clusters.

Building on the two principles mentioned at the beginning of this section, we conceived the framework's goals to be to solve three tasks in the realms of ML model performance analysis, which are stated and defined as follows:

1. **Comparison**: Being able to compare the performance of two or more ML models.
2. **Evaluation**: Being able to assess a single model's performance.
3. **Understanding**: Being able to identify potential causal relationships between a model's input data qualities and its performance.

1.3 Research Questions

These three tasks, together with our wish to build on and improve analysis with classical evaluation metrics, have inspired the following Research Question (RQ)s. The primary research question we aim to answer with our work is:

- **RQ** - "Can we use a data type-, task- and model-agnostic, clustering-based framework to compare, evaluate and understand model performance in a machine learning problem?".

Our Secondary Research Question (SRQ)s dissect the primary research question into more isolated components, while also comparing the framework to classical evaluation metrics:

- **SRQ 1** - "Can we use a data type-, task- and model-agnostic, clustering-based framework to compare model performance in a machine learning problem?"
- **SRQ 2** - "Can we use a data type-, task- and model-agnostic, clustering-based framework to evaluate model performance in a machine learning problem?"
- **SRQ 3** - "Can we use a data type-, task- and model-agnostic, clustering-based framework to understand model performance in a machine learning problem?"
- **SRQ 4** - "Does a data type-, task- and model-agnostic, clustering-based framework give more insight when comparing, evaluating and understanding model performance in a machine learning problem than classical evaluation metrics?"

To perform qualitative evaluation towards answering the research questions, the ClAIrify's guidelines will be followed to implement a practical application, applied to two use cases, and evaluated through a think aloud protocol. Quantitative evaluation of the study will be achieved through the participants from the think aloud protocol study answering a post-study, Likert-scale questionnaire. This approach is further explained in section 4.1.

1.4 Contribution Statement

The main contributions of this Masters' thesis are:

- Proposal of a novel conceptual visual framework based on clustering that allows for more interpretability during evaluation of multiple ML models while remaining data-, task-, and model-agnostic.
- Validate the conceptual framework through an instantiation, tested through a Think Aloud Protocol (TAP) study, a survey and two use case studies.

1.5 Report Structure

The remainder of this report is structured as follows. Firstly, chapter 2 gives an overview of other related works to this one, framing our work within past research. chapter 3 then goes on to explain the conceptual framework, and its implementation. The experiment's setup and results are then reported on, and evaluated in chapter 4. chapter 5 talks about our approach's limitations and possible future research lines. Finally, our research's fruits are given a conclusion in chapter 6

2 Related work

Due to the broad scope of our framework, it is imperative to frame our approach within the research fields it is part of. To this end, in this chapter, we will zoom in from the framework's characterisation within the field of VA and ML, through to its place in model interpretability, to finally compare it to similar frameworks, some of which served as inspiration for this research.

2.1 Within Visual Analytics & Machine Learning

Within the overlap of VA and ML, [36] propose three sub-fields depending on which phase of the ML pipeline the VA technique is applied in. These fields are:

1. *Understanding* - why models behave the way they do
2. *Diagnosis* - model training performance analysis
3. *Refinement* - improve a model or make it more robust.

Our framework falls into the second category, as ClAIrify serves as a *diagnostic* tool to evaluate model performance. It does not fall into the *understanding* category due to its model-agnostic nature. While using ClAIrify combined with domain knowledge can serve as support for *refinement*, it was not specifically designed to do so.

Furthermore, as a tool focusing on the power of visualisation, it is also necessary to identify our research within the field of VA. To this end, we use the six question proposed by [27] which they envisioned to categorise VA techniques. We answer the questions according to the article's format/options as follows:

- **WHY** use our framework? For explainability and comparing & selecting models.

- **WHO** uses the framework? Model developers & builders.

- **WHAT** is visualised? Aggregated information from various processed sources from ML models' inputs and outputs.

- **HOW** is it visualised? Dimensionality reduction, scatter plots and heatmaps.

- **WHEN** is it used? During evaluation.

- **WHERE** is it used? This question is split into two sub-questions:
  - Which application domain does this work fall into? General model performance evaluation.
  - Where has the research been conducted? Utrecht University.

[16] surveys the contemporary research advances in using VA for ML understanding. They highlight the necessity of balancing human and machine effort in regards to VA systems. This is also a concern in this project, where we aim to take some load off the framework in order to give more freedom to the user. This is so as the user is responsible for both the model training and the choice of an item-wise performance metric which affects the visualisations. This is explained further in chapter 3. We believe this potentially enhances the explainability of the framework under the assumption that either or both the choice of model and metric are made following prior knowledge of the developer.

2.2 Machine Learning Interpretability

Interpretability has been a topic with a surge in research in recent times. We designed our framework to serve the purpose of more effectively interpreting ML model performance, and thus, it falls into this field.

Interpretability is a broad research field with many different approaches, and we situate ClAIrify within it following the characterisation of the domain introduced in [33].

They identify three opportunities in which interpretability is desirable: (1) Data understanding & discovery, (2) Trust building and accountability and (3) Model comparison & diagnostics.

Moreover, they identify two generic approaches in model interpretation with visual analytics: (a) Visualising model structure (White-Box), and (b) Visualising model behaviour (Black-Box)

Within this scheme, our approach aligns mostly with the (1) data understanding & discovery and (3) model comparison & diagnostic fields. Secondly, due to ClAIrify being model-agnostic, our approach for model interpretation with VA is (b) visualising model behaviour.

Furthermore, they address a common claim that was taken into account for the design of our framework. ML techniques are often black-boxes lacking in inherent interpretability. Making a model more interpretable is commonly believed to come with a sacrifice in performance [9]. Nonetheless, they prove that analysing the input-output relationships of models can provide a valuable means of increasing the interpretability of the predictions without affecting performance. This is one of the main reasons behind using a model-agnostic approach in ClAIrify.

2.3 Similar Frameworks

Typically, approaches to interpret ML models take a white-box approach, accessing the internal architecture of the model itself [31, 37, 43, 44, 51, 64, 66]. Even though ClAIrify doesn't follow this approach, we have taken inspiration from *RegressionExplorer*, presented in [15]. It is a model-specific tool for interactive exploration of Linear Regression (LR) models. Contrary to ours, it allows users to generate new models based on previous analysis offered by the application. Moreover, it enables a performance comparison across user-defined sub-populations. We also work with sub-populations, but use clustering instead, to automatically generate them, with the purpose of potentially taking some cognitive load off of the user.

Nonetheless, ClAIrify is a framework for analysis of model behaviour, i.e., analysis of the input-output relationships of the models. This is often considered to be a black-box approach. Thus, we focus more on these kind of works in this section.

One of the main reasons for using a visual interface for analysing performance of machine learning models is that these offer much more fine-grained details and interactivities than classical metrics, which normally aggregate performance, giving a too general view. The goal of our framework is to leverage such summary statistics to be able to zoom in from coarse-grained performance anomalies to fine-grained potential causes.

There have been other approaches to provide exploration tools at the fine-grained level, such as the Confusion Wheel presented by [2] that proposes an enhanced equivalent of a confusion matrix for multi-class classification that displays the ratios of true/false positives/negatives per class in a circular layout. The downside of this approach is that compared to ours, it does not allow for comparison between models.

Our other main inspiration, *Manifold*, presented in [68], is a framework for the interpretation and diagnosis of ML models. It's also model agnostic, as it only requires the models' input and output to work. It includes visual summaries of pairwise model agreement or disagreement, which can help when deciding between a specific pair of models. This is useful, but doesn't allow for a simple global comparison between models, which is what we are designing our framework for. This potentially makes it faster to detect and zoom into a symptomatic set of data instances and models.

As for VA frameworks and techniques to understand models during different stages of the ML workflow, there have been many approaches [15, 17, 34, 35, 56, 67, 68]. Many of them fall into [36]'s *model understanding* category. But others, like ExplAIner, the framework proposed in [56], are an inspiration for us. Like ours, it's a conceptual framework instantiated with a VA system, but while we only try to be able to diagnose and compare models, they allow for visual analysis in every step of the ML workflow, and also target model novices by integrating educational information. Meanwhile, we focus on one single iteration of diagnosis and/or refinement, adding clustering as a key step for understanding each model's performance on the data.

3 Methodology

In this section, we will explain the conceptual framework, its appropriate data pipeline, as well as the specific implementation.

3.1 Conceptual Framework & Data Pipeline

Our conceptual framework, ClAIrify, provides a recipe for the abstract building blocks that should be part of any of its implementations. With the goal of the framework being to create visual model performance profiles over the input data, and having in mind the model-, data-type- and task-agnostic qualities, we first had to set up a threshold. Its purpose was to balance the amount of work done by the user to prepare the data, versus the amount of load sustained by the framework itself to provide the necessary visualisations to support analysis.

This load balance was set based on the restrictions imposed implicitly according to the three aforementioned agnostic qualities, which will be explained after presenting the necessary input to the framework. The only necessary input to the framework is two-fold:

- *Evaluation data*: The data that the models were evaluated on. This doesn't per se make reference to the evaluation set typically used in machine learning practice to improve generalisation. It can be any set of data points that the user deems worthy of performance analysis.
- *Instance-wise error-score per model*: For each model and for each instance of the evaluation data, an error metric shall be calculated, assigned, and input into the framework. Here, we give the user the choice of error metric, enabling them to use one of the classical evaluation metrics (e.g. Root Mean Square Error, Mean Average Percentage Error), or any other metric that meets their interpretability criteria.

Model-agnosticity entails that the framework should have no knowledge of models' inner logic. Many approaches to explainability and performance analysis in the context of ML focus on model-specific techniques. These give insights relying on the knowledge of the inner logic of a model. But, as *generality* is one of the main attributes of this framework, we chose a model-agnostic approach, considering models black-boxes. The user must only introduce the evaluation data and the instance-wise error-score of each model that they chose. This allows us to shift the focus to the evaluation data instances, making it a data-centric approach. We provide tools to find out which qualities of the evaluation data are potentially affecting the models' performance.

An approach could have been to design ClAIrify to train the models chosen by the users. This would allow any type of models to be input into the framework, setting it up in such a way to interact with a wrapper from the ClAIrify framework to enable training and training-parameterisation altogether. We didn't choose this option, as it would imply loosing control over each model's training, as the framework's wrapper would have to generalise functionality to be able to train any kind of models. To keep in line with our data-centric approach, we chose to give the user full control of data preprocessing and model training by leaving these tasks in their hands, and only requiring the knowledge of an error score based on each trained model's evaluation predictions.

Machine learning task-agnosticity (e.g. classification or regression) means that the framework should allow for compatibility with any tasks in the field of machine learning. This comes with a certain limitation, i.e., the impossibility to enforce full agnosticity in this regard, because of the multiple ways of evaluating different machine learning problems. In our case, the tasks need to fulfil the singular condition of having *multiple evaluation items*. This means each model should be tested on more than one evaluation instance. An example of what would be compatible with ClAIrify is a reinforcement learning task that tries to find an agent that performs the best in a video-game. The evaluation data could be the score of several reinforcement learning architectures (models) on multiple levels in the video game (evaluation data), according to a certain user-defined metric (e.g. time to complete a level). Following along with the previous example, a task that would not be analysed as well with ClAIrify is finding an agent that performs well on a single level. Even if there are multiple agents tested, there is only a single instance of evaluation data, rendering clustering useless.

Lastly, data-type-agnosticity implies that the evaluation data input into the framework should be able to be of any type desired by the user, be it images, time series, or else. An important clarification has to be made here. The framework is data-type-agnostic only in its conceptual form, as the visualisation guidelines designed for it work with any type of data. This agnosticity is lost when it comes to the implementations, as different types of data require different visualisations. Not only that, but visualisations should also be context dependent [53] for increased effectiveness, which means that they are likely to depend on the domain of the machine learning problem. The guidelines to implement these visualisation components will be explained later on in this section.

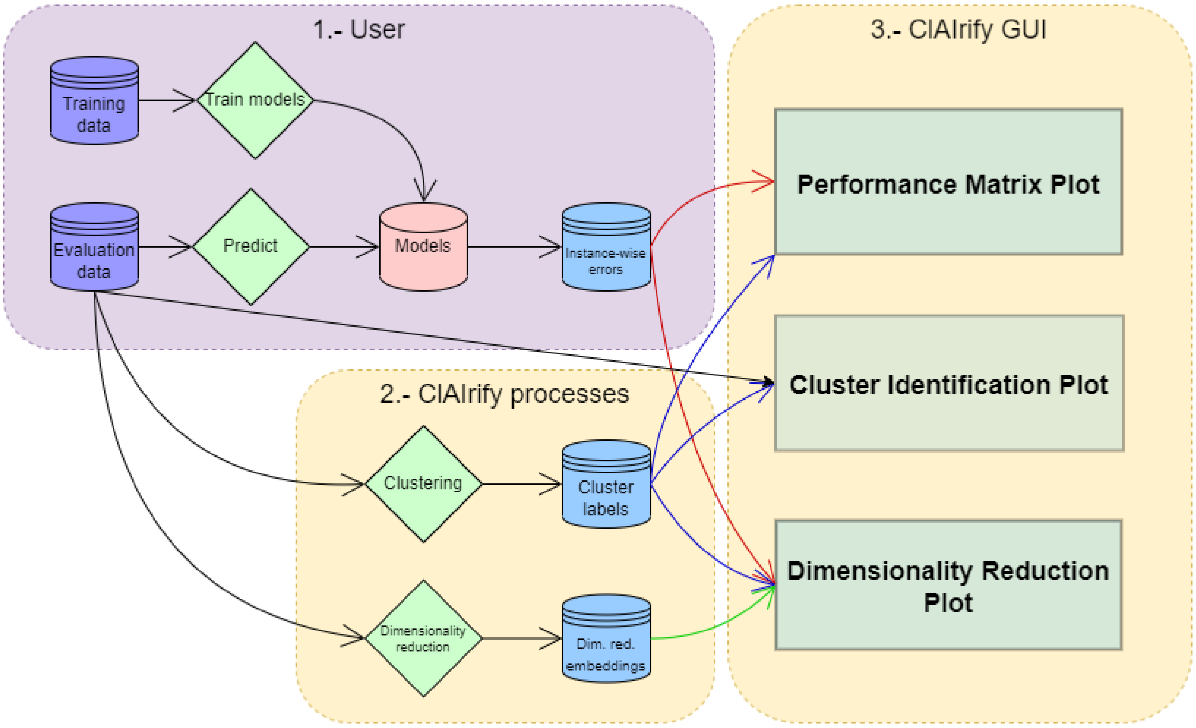


Figure 3.1: Diagram of the data pipeline representing each step that the user must take, as well as each component of the framework, which is itself divided into segments (1 to 3) belonging to the back-end processes and the Graphic User Interface. The lines between the components show the data flow between the processes/components.

Figure 3.1 shows the data pipeline, with the steps the user must perform as well as all of our framework's components, and how the data flows between these. What follows is the explanation of each numbered segment of the pipeline according to this figure.

Part 1 of the pipeline corresponds to the user's work load. First of all, they must obtain *training data* and *evaluation data*. As hinted at earlier, these may well overlap in content, if the user deems it necessary for analysis; the distinction is made just because they are used for different purposes in the pipeline. After this, the user must choose the types of models for which they want to create performance profiles over the gathered data. These models may be any kind of model that is suitable for the collected data. They can differ in facets such as architecture, initialisation parameters, or training parameters. Once selected, the models are trained on the *training data*, and these trained models are used to make predictions on the evaluation data. During this last step, the user must choose the *instance-wise-evaluation metric* they want to use to continue analysis with our framework. With the model's predictions on the evaluation data, the user must generate a file containing the error score for each instance and for each model. This may consist of a table matching error scores to instances and models. This is where the work load is taken off the user, who will now introduce both the evaluation data and the instance-wise-error file into the framework to start analysis.

Part 2 of the pipeline corresponds to the processes that the ClAIrify framework is in charge of that don't have any direct visual output. Still, they are the necessary building blocks to produce the information necessary for the Graphic User Interface (GUI). Firstly, there is *clustering*. As mentioned before its a key point of the ClAIrify framework. Here, the implementer of the framework must choose a clustering algorithm (e.g., k-means, k-medians) that they deem appropriate to divide the evaluation data into heterogeneous groups. It is so important because it is the cornerstone of understanding. A solid choice here will potentially create meaningful clusters to which the user can attribute semantic meaning. This algorithm is applied on the evaluation data to come up with the *cluster labels* used as input for the GUI. Secondly, there is *dimensionality reduction*. It is used to morph high-dimensional data into 2-dimensional data. Its purpose is mainly to help with visualisation and interactivity. This technique places similar instances close together and thus potentially can aid in more fine grained analysis, as will be seen later on. Here too, an appropriate technique (e.g., Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE)) shall be chosen, so that the processed evaluation data can be displayed in the next and last part of ClAIrify.

This last section corresponds to the ClAIrify Graphic User Interface. The general guideline for this user interface is that it should preferably fit all its components in a single screen, to ensure facilitating a global overview. Also, it should be designed to be displayed on a moderately big surface, to allow for detailed view of its components, for example by using a PC/laptop monitor. As mentioned, there are three visualisation components in our framework, which may be complemented with further information and cross-filtering, as deemed necessary. The following list details the components and their design guidelines:

- *performance matrix plot*: This plot should be a matrix-like diagram showing the average performance of each model on each cluster. Colour-coding each cell of the matrix to help analysis is highly recommended. This plot is envisioned to be the central piece of the framework, the starting point of all error-tracing. It provides the most general view of performance on each cluster, allowing users to then use the other components to give a meaning to the data and to dive deeper into the potential causes of performance issues in this cluster.
- *cluster identification plot*: This plot is a key component to semantic explainability of the clusters. It should, using the information from the cluster labels obtained in the clustering step, offer a simplified (preferably one graph per cluster) representation of the data points in each cluster. This can be achieved in many ways, where the main goal should be optimal synthesis of information for explanation. An example for multiple-feature input could be to show a violin plot of all the features per cluster. On the other side, if there is only a single feature, then there could be a histogram of the values that feature takes for that cluster.
- *dimensionality reduction plot*: this plot should take the output of the dimensionality

reduction process and use it to show a 2D scatter-plot. Furthermore, it should implement interactivity to be able to select points and then inspect their features, in a similar way to the cluster identification plot. This visualisation is to be seen as the interface for the most fine-grained analysis, allowing the user to "zoom in" on particular instances, because they are potentially problematic, interesting, or for any other reason. To embed this plot in the greater scheme of the application and make its function symbiotic with the other plots, there are other two requirements this plot must accomplish.

1. Firstly, there must be a visual cue as to which cluster each point belongs to, which can be done by using a colour-scheme that is consistent with colours used in the cluster identification plot.
2. Secondly, there has to be another cue as to how well each point performed on average, paired with the possibility to filter for performance per model. This cue could use a colour-scheme consistent with the one used in the performance matrix plot.

It may be visually burdensome to display two colours per point to comply with both of the requirements. Thus, it is suggested to avoid this, and maybe instead signal cluster membership by plotting the convex hulls around the points of each cluster, using the cluster's colour.

These are the basic guidelines for the building blocks of our proposed three-way-agnostic framework. Finally, it is suggested that each implementation makes the design choices to add extra cross-filtering to the interface that are deemed adequate.

3.2 Algorithmic Implementation

According to the actionable guidelines presented in the previous section, and within the scope of this project, all three steps of the data pipeline were implemented. The framework was implemented to work on univariate time series data. The first step is more relevant for the use cases, and thus will be exemplified in section 4.4. This section focuses on the instantiation of the back-end processes of ClAIrify, i.e. the 2$^{\text{nd}}$ segment of Figure 3.1. The framework was implemented using the Python programming language, in its 3$^{\text{rd}}$ version [63].

As stipulated in the previous section, the framework loads two files, one with the evaluation data, and one with the instance-wise error scores. Both of these must be saved as a Panda's DataFrame [60] in Python's standard Pickle format [63] to be loaded correctly. The specific structure of each of these files is as follows:

- Each row of the *evaluation data* file should consist of one of the data instances (i.e. time series) and each column of the DataFrame must correspond to one time-step.

- The rows of the *instance-wise error-score* file must also be corresponding to individual data instances, consistent with the evaluation data file. Each column, on the other hand, shall correspond to each model that is under scrutiny.

Once these have been specified, the program loads the files and uses them to carry out the processes present in block 2 of the pipeline. The clustering algorithm that was chosen was k-means, as it is time efficient, and, despite its simplicity, a popular clustering algorithm widely used in the literature [65]. It was implemented using the tslearn Python library [59], a popular library for working with time series data. Specifically, the TimeSeriesKMeans algorithm was used, which implements clustering algorithm for this data type. The distance measure used as similarity criteria in clustering was Dynamic Time Warping (DTW) [46], as it takes into account the nature of the time series data and is suitable for comparing time series that have similar patterns occurring at different speeds. A widely known limitation of the k-means algorithm is the k-problem, which speaks of the necessity of configuring said clustering algorithm with a k number of final clusters we wish to obtain. This assumes that the adequate number of clusters is known beforehand, which is normally not the case, and potentially less so in the case of this specific framework, which is to be used to find relationships between the models' performance and the underlying patterns of the data. The premise of the framework and the assumption of the k-means clustering algorithm are colliding, and that is why we chose to implement a variation of the popular rule of thumb for deciding k. [32] states that as a rule of thumb, k should be:

$$k = \sqrt{\frac{n}{2}}$$

Here n is the number of instances in the data set. Our variation on the other hand, takes into account the limited space available on a computer screen (i.e. the preferred medium of display for our web app), and thus sets a limitation to this amount of clusters, so that it be a maximum of 5, in accordance with the empirical realisations based on the test runs of the app. This leads to the equation for k in our framework being:

$$k = \min(5, \sqrt{\frac{n}{2}})$$

To further enhance the quality of clustering, we use tslearn's argument options to set the k-means algorithm to be run 10 times with different initialisations, to finally keep the best option in terms of inertia. Inertia, or the within-cluster sum of squared distances is an optimisation criterion used to evaluate an algorithm's quality in clustering. It is a measure of coherence of each cluster. The formula for calculating it is as follows:

$$\text{inertia} = \sum_{i=1}^{N} |x_i - C_i|$$

So, it is the sum of the absolute distance of the N data samples x_i with respect to the cluster centre of the cluster they belong to C_i. Other versions implement it as the squared distance, instead of the absolute distance, but in this case the latter is preferred in order to avoid overly big values in likely high-dimensional data. The assumption is that a good clustering result is one with a low inertia value.

In the case of dimensionality reduction, a more complex approach was chosen. In this case, a total of three different dimensionality reduction algorithms are run on the evaluation data, which are the following:

- *PCA* - As introduced in [30], Principal Component Analysis (PCA) aims to analyse inter-correlated quantitative dependent variables in order to come up with a set of new variables orthogonal to each other, which are called principal components. Our implementation uses the PCA version of the scikit-learn Python library [48], parameterising it to generate two principal components out of the data.
- *t-SNE* - This version of the Stochastic Neighbour Embedding, t-SNE is an algorithm to visualise high-dimensional data that reduces the tendency to crowd points together in the centre of the map, as described in [62]. The implementation we used was the one found in the scikit-learn library [48] too.
- *Uniform Manifold Approximation and Projection (UMAP)* - Presented in [42], the Uniform Manifold Approximation and Projection for dimension reduction (UMAP) algorithm is based on Riemannian geometry and algebraic topology and is considered to preserve more of the global structure of data, as well as have a superior run time performance than t-SNE. We used the implementation offered by the namesake *umap* Python library.

The reason behind three different algorithms being run is twofold. Firstly, it gives the user the option to choose between three visualisations based on algorithms commonly used in the literature. Secondly, these different visualisation options give the user the choice to use the visualisation that suits the user the most for their analysis purposes. This visual suitability will be discussed later on.

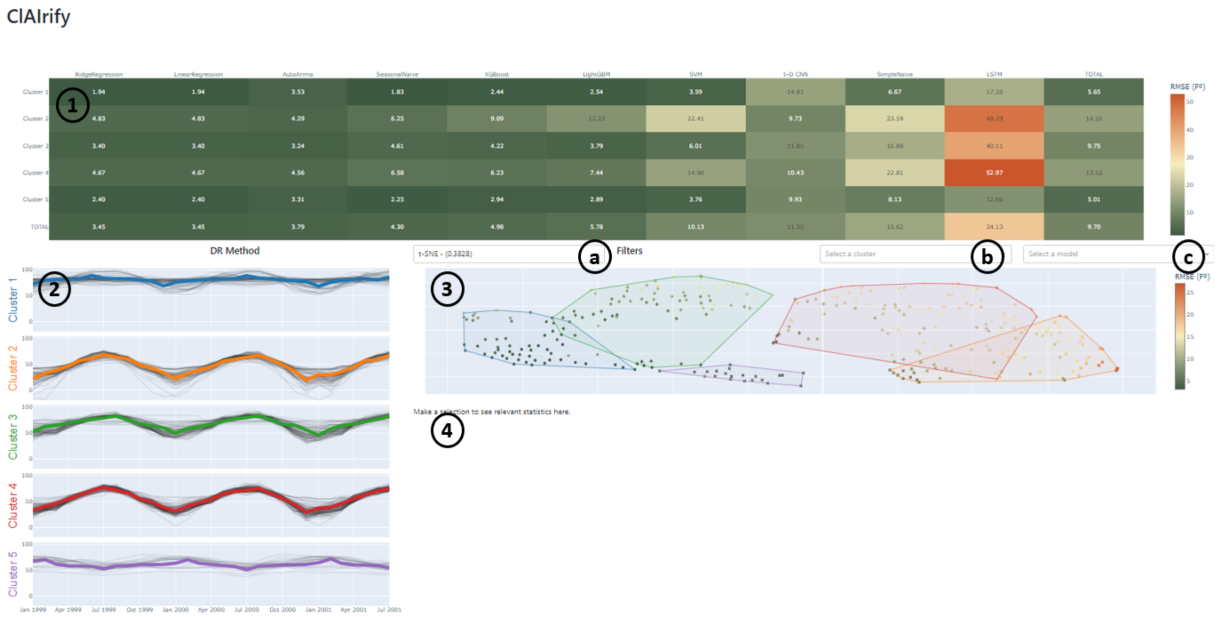
Moreover, for each of these algorithms, the silhouette score is calculated using the cluster labels and the reduced 2D instances as input. Envisioned in [52], the silhouette score says something about the clustering quality, and takes real values in the range of [-1, 1]. Each data point gets assigned a silhouette coefficient, also in the range [-1, 1], which can be interpreted as follows:

- A coefficient closer to 1 means the point is similar to other data objects in its same cluster, while being dissimilar to ones outside of it.
- At coefficient 0, the point is close to a decision boundary between clusters.
- If the coefficient, on the other hand, is closer to -1, this means a data point is far away from data points in its assigned cluster and closer to points in another cluster , implying potential misclassification.

The silhouette score is simply the average of all the points' silhouette coefficients. This score is then used to give the user an intuition about which dimensionality reduction algorithm gave the potentially higher quality results.

The product of the clustering and the dimensionality reduction are a set of cluster labels and three sets of 2D points (one for each dimensionality reduction algorithm). For further processing, each row of these four files (*evaluation data*, *instance-wise error-score*, *cluster labels* & *dimensionality reduction embeddings*) corresponds to the information related to the data instance in the same row of the evaluation data. This consistency facilitates further processing and ensures consistency in the visualisations.

3.3 Graphic User Interface



ClAIrify

	RidgeRegression	LinearRegression	AutoArima	SeasonalNaive	XGBoost	LightGBM	SVM	1-D CNN	SimpleNaive	LSTM	TOTAL
Cluster 1	1.94	1.94	3.53	1.83	2.44	2.54	3.59	14.81	6.67	17.20	5.65
Cluster 2	4.83	4.83	4.29	6.25	9.09	12.23	22.41	9.73	23.59	48.29	14.55
Cluster 3	3.40	3.40	3.24	4.61	4.22	3.79	6.01	11.85	16.89	40.11	9.75
Cluster 4	4.67	4.67	4.56	6.58	6.23	7.44	14.90	10.43	22.81	52.97	13.52
Cluster 5	2.40	2.40	3.31	2.25	2.94	2.89	3.76	9.93	8.13	12.06	5.01
TOTAL	3.45	3.45	3.79	4.30	4.98	5.78	10.13	11.35	15.62	34.13	9.70

DR Method: t-SNE - (0.3828)

Filters: Select a cluster; Select a model

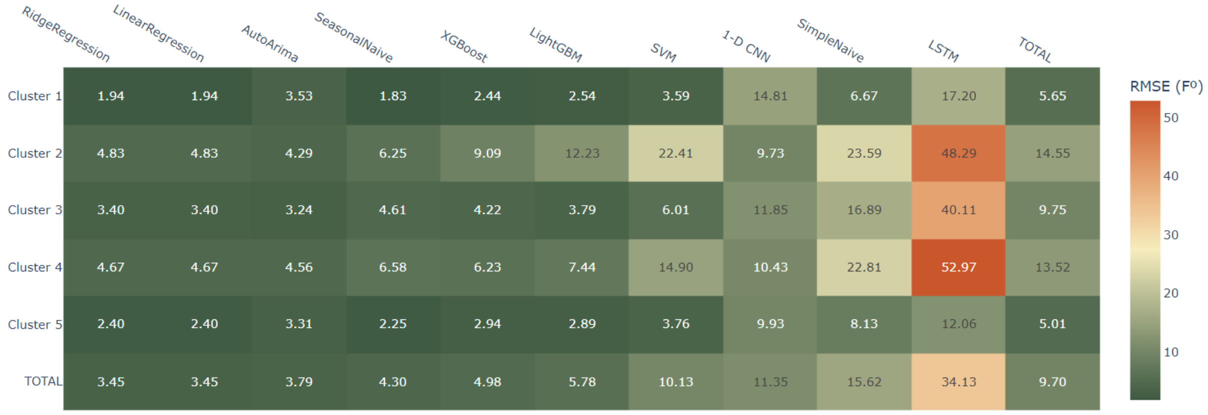
Make a selection to see relevant statistics here.

Figure 3.2: ClAIrify dashboard, displaying the numbered visualisations information outputs (1) cluster-model performance matrix, (2) cluster identification plot, (3) dimensionality reduction plot & (4) selection summary display, as well as the filtering drop downs to choose the (a) dimensionality reduction technique, (b) selected cluster and the (c) selected model.

Now I will describe the functionality of the GUI, which makes use of the data acquired in all the previous stages. It corresponds to the 3rd segment of Figure 3.1. The ClAIrify dashboard, accessed as a web app, can be seen in Figure 3.2. Any circled numbers or letter used from now on in the text will make reference to the ones seen in this figure.

ClAIrify runs as a web app on localhost, thanks to the functionality of the Dash framework [11], which was chosen because of its powerful cross-filtering functionalities and its seamless integration with the Plotly plotting library [28], the cornerstone for displaying all of our visualisations.

This interface follows the recipe given by the guidelines presented in section 3.1. It contains a ① *performance matrix plot*, ② a *cluster identification plot* and a ③ *dimensionality reduction plot*. These will be further explained in the next three sections. Apart from instantiating these three main components of the framework, there is also a ④ *selection summary display* which was added with the purpose of further enhancing the analysis and shows basic summary statistics. The last subsection concerns the cross-filtering capabilities of the implementation.



	RidgeRegression	LinearRegression	AutoArima	SeasonalNaive	XGBoost	LightGBM	SVM	1-D CNN	SimpleNaive	LSTM	TOTAL
Cluster 1	1.94	1.94	3.53	1.83	2.44	2.54	3.59	14.81	6.67	17.20	5.65
Cluster 2	4.83	4.83	4.29	6.25	9.09	12.23	22.41	9.73	23.59	48.29	14.55
Cluster 3	3.40	3.40	3.24	4.61	4.22	3.79	6.01	11.85	16.89	40.11	9.75
Cluster 4	4.67	4.67	4.56	6.58	6.23	7.44	14.90	10.43	22.81	52.97	13.52
Cluster 5	2.40	2.40	3.31	2.25	2.94	2.89	3.76	9.93	8.13	12.06	5.01
TOTAL	3.45	3.45	3.79	4.30	4.98	5.78	10.13	11.35	15.62	34.13	9.70

Figure 3.3: Performance matrix plot showing the average performance of each model on each cluster.

3.3.1 Performance Matrix Plot

Figure 3.3 displays the ① *performance matrix plot*, implemented as a heat-map matrix, which summarises the general performance of each model on each cluster. Apart from the colour-coding used to visually enhance the performance, each cell also contains the numeric value of the average of the error score of each model on a specific cluster. There are other two additions to the general guidelines. Firstly, there is a colour bar indicating which colour corresponds to what performance score, aiding intuition about how to interpret each colour. Secondly, there is an extra column and an extra row appended to the right and bottom of the heat-map, respectively. These show the marginal averages, and the average overall performance on a specific cluster and model, respectively. Furthermore, the bottom-right cell indicates the average performance of all models on all the data. This was implemented to easily detect problematic clusters or models. Lastly, the columns, i.e. the models are sorted left to right from highest to lowest average performance on all the clusters, again to lower the overall cognitive load on the user.

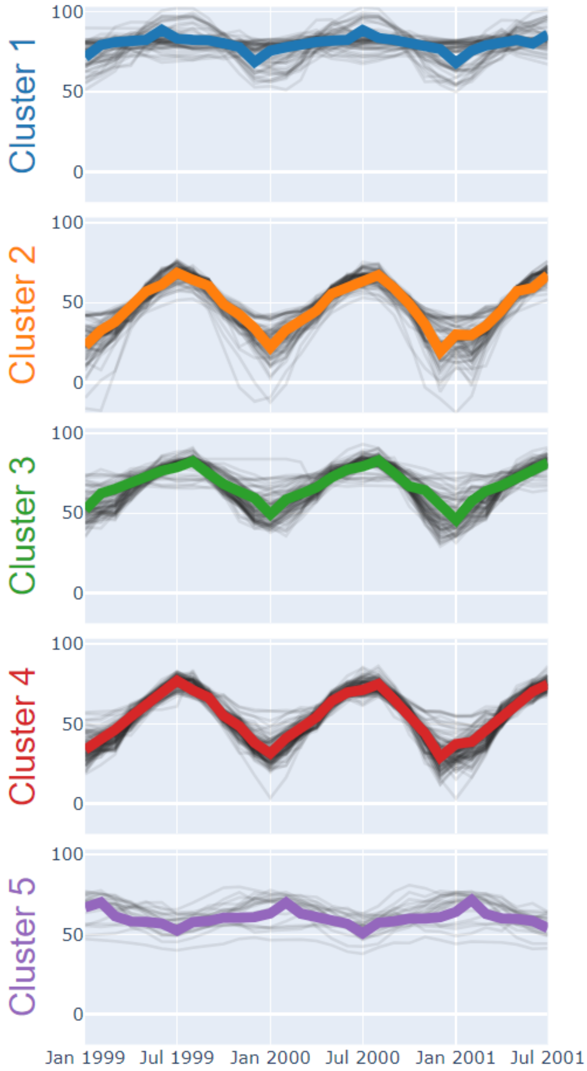


Figure 3.4: Cluster identification plot showcasing 5 clusters of temperature time series data.

3.3.2 Cluster Identification Plot

Figure 3.4 showcases the ② *cluster identification plot*, key to give semantic sense to the data. It contains one subplot for each cluster. In this case, as the evaluation data consisted of time series, this specific display was chosen. In it, each individual time series in the cluster is plotted as a low opacity grey line-plot. This already gives a sense of the cluster, as the darker-grey areas imply more lines of the cluster being present in the area, and may suggest a trend. On top of that, we added an identifier for each subplot, which represents the dynamic time warping barycenter average of all the time series contained in each cluster. The barycenter average was calculated using the *dtw_barycenter_averaging_subgradient* function of the *tslearn* Python library, which implements the averaging method, using the stochastic subgradient descent method, which is proven to give better results in faster time than expectation-maximisation, another way of finding the average [54]. It is out of this scope to explain the details of this algorithm, but in essence, the barycenter is a time series b, is the time series that accomplishes the following function, given a set of data x:

$$\min \sum_i d(b.x_i)^2$$

Here, d is the distance function, which is the dynamic time warping metric, also used previously in k-means. This minimisation is an optimisation problem, solved with the mentioned stochastic subgradient descent. This identifier is meant to be another visual aid to give the average shape of the time series in the cluster. Both the cluster identifier and subplot label are given a colour, which is used to for identification purposes in the GUI.

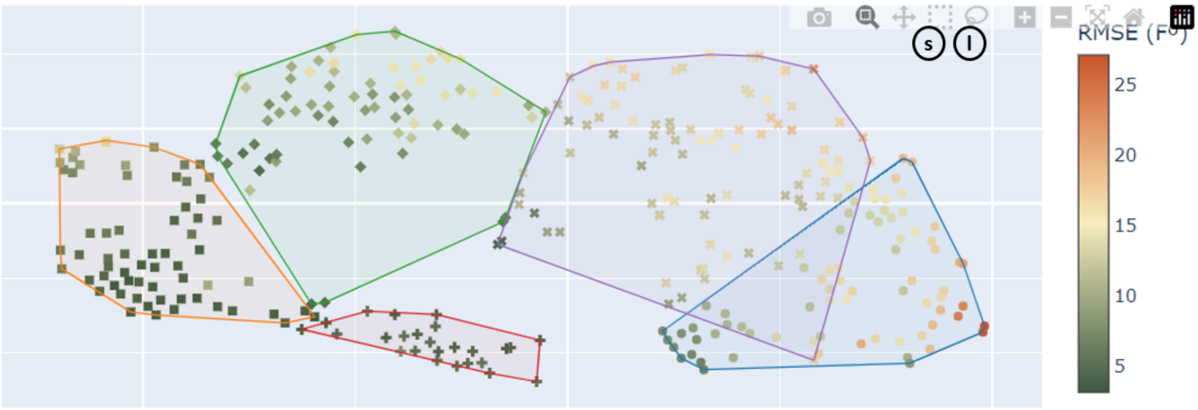
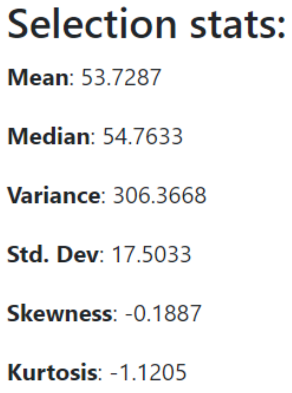


Figure 3.5: Dimensionality reduction scatter-plot showcasing the colour-coded average performance. Marked with encircled letters are the (s) square selection and the (l) lasso selection tools.

3.3.3 Dimensionality Reduction Plot

Figure 3.5 shows the ③ *dimensionality reduction plot*, which consists of a scatter plot of the 2D dimensionality reduced evaluation data instances. By default, the dimension embeddings displayed are the ones obtained by the dimensionality reduction algorithm with the highest silhouette score. There are several design choices that link these points to the other two graphs. For instance, each point is plotted in a certain colour, according to the same colour scheme used in the *performance matrix plot*. This is so for easy identification of the more problematic data instances. Moreover, each point's cluster assignment is shown in two different ways. On one hand, the polygons, which are the convex hulls of each cluster, are plotted in the colour of the respective cluster. Nonetheless, it can be the case that certain clusters overlap, as is the case with the purple and the blue clusters in Figure 3.5. Thus, we chose to give each individual point a different shape, to be able to discern different cluster's points on closer inspection. Here, *Dash*'s integrated tooltip also aids in identification. Figure 3.5 also includes the toolbar, displayed on the top right. Even though all three of the graphs include this toolbar, this one has the most filtering interactivity attached to it, focused on the two tools highlighted by the encircled letter: the ⓢ square selection tool and the ⓛ lasso selection tool.



Selection stats:

Mean: 53.7287

Median: 54.7633

Variance: 306.3668

Std. Dev: 17.5033

Skewness: -0.1887

Kurtosis: -1.1205

Figure 3.6: View of the *selection summary display* once the user makes a selection of a subgroup of data points.

3.3.4 Cross-Filtering

Filtering is a big part of the ClAIrify framework, as it permits the user to zoom in on sets of data points that they deem worthy of further inspection. Both in Figures 3.2 and 3.5, the encircled letters identify the filtering options at the user's disposal.

When filtering is applied to select a subgroup of the data, the ④ *selection summary display* changes from how it appears in Figure 3.2, which is the default appearance, to how it appears in Figure 3.6. The selection prompts the framework to display a few a number of widely used statistics about the selected data points: mean, median, variance, standard deviation, skewness and kurtosis. These are complementary aids to understand the data that was filtered.

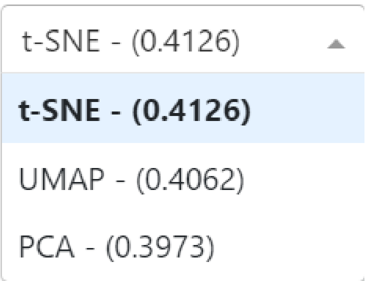


Figure 3.7: Dimensionality reduction dropdown displaying available methods and their silhouette scores in parentheses.

The implemented ClAIrify GUI contains three dropdown widgets to filter the data and modify the visual output of the dashboard. The ⓐ dimensionality reduction dropdown in Figure 3.7 shows the user the three dimensionality reduction techniques available, together with their respectively obtained silhouette score in parentheses. By default the highest scoring technique's 2D embeddings are displayed, and the dropdown is sorted accordingly, from the highest to the lowest silhouette score. This allows the user to choose the embedding that they are more comfortable with for analysis.

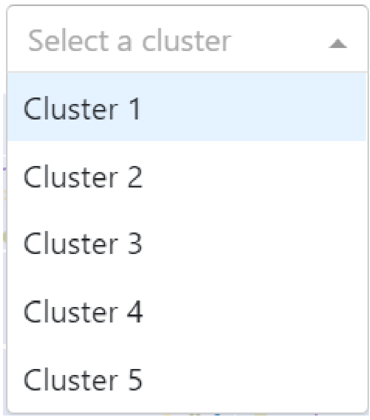
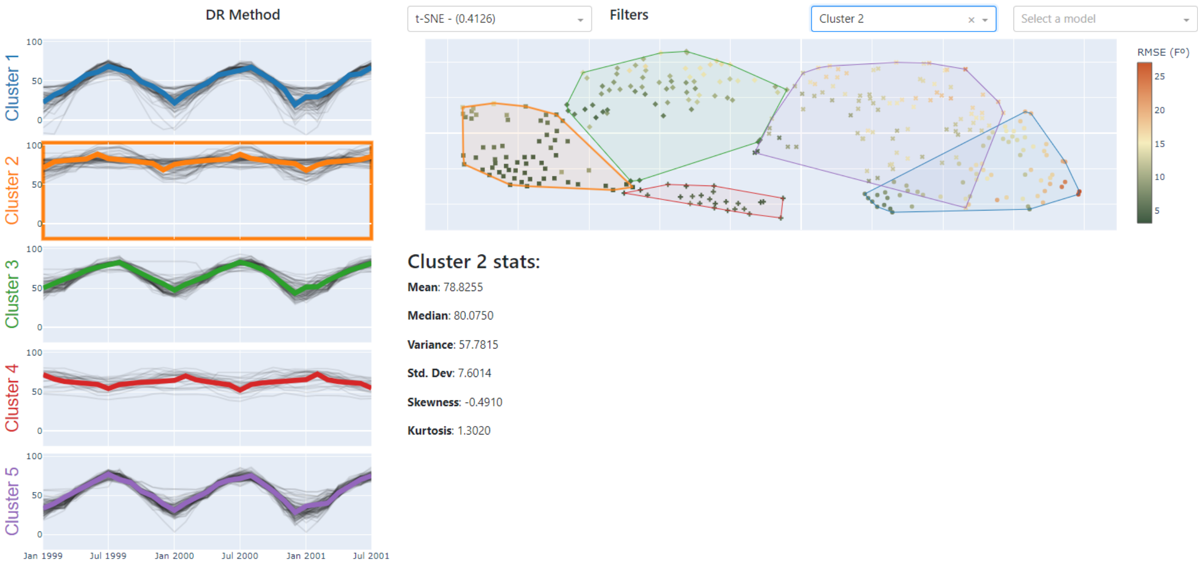


Figure 3.8: Cluster selection dropdown displaying all available clusters.



Cluster 2 stats:

Mean: 78.8255

Median: 80.0750

Variance: 57.7815

Std. Dev: 7.6014

Skewness: -0.4910

Kurtosis: 1.3020

Figure 3.9: Modified dashboard after selecting a cluster from the cluster selection dropdown.

Figure 3.8 shows the ⓑ *cluster selection dropdown*, which allows to select one of the found clusters. This action modifies both the ② *cluster identification plot* and the ③ *dimensionality reduction plot* to highlight the selected cluster, for visual support. In the former, the cluster's subplot's borders are highlighted in the clusters colour, while in the latter, the convex hull is highlighted. Furthermore, a cluster selection modifies the ④ *selection summary display* to display statistics of the picked cluster. These visual modifications can be observed in Figure 3.9.

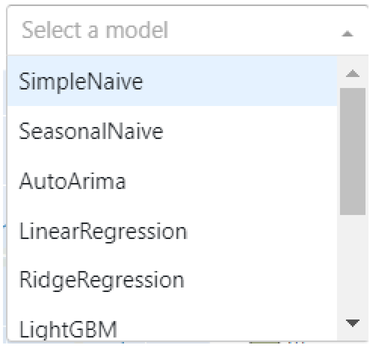
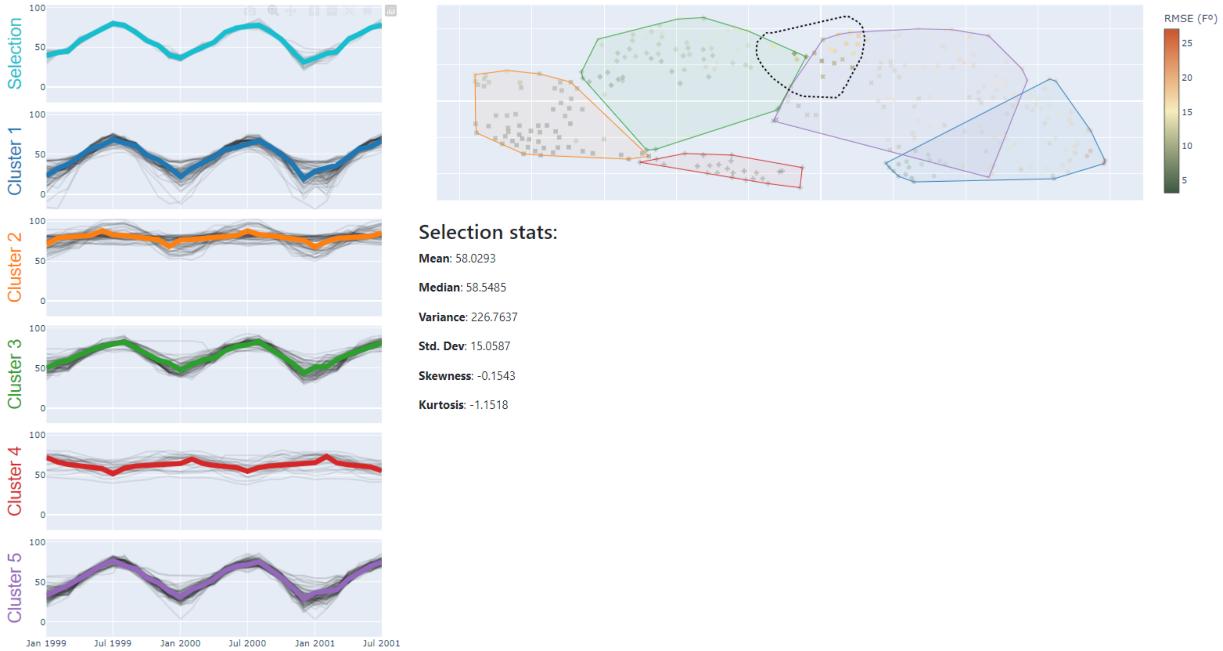


Figure 3.10: Model selection dropdown displaying all available models for filtering.

The ⓒ model selection dropdown shown in Figure 3.10 allows the user to select a specific model, from the ones available in the error score data. This selection only modifies the ⓒ *dimensionality reduction plot*, by changing the colours of the points and the colour bar. The points will display the colour according to the error score obtained by the selected model. The colour bar will be modified to display the new range of error score values of the said model.

Both the ⓑ cluster selection dropdown and the ⓒ model selection dropdown allow deselection, which reverses the mentioned visual changes.



Selection stats:

Mean: 58.0293

Median: 58.5485

Variance: 226.7637

Std. Dev: 15.0587

Skewness: -0.1543

Kurtosis: -1.1518

Figure 3.11: Modified dashboard after selecting a subgroup of points from the dimensionality reduction plot.

Manual filtering can be done with the tools shown in the ③ *dimensionality reduction plot*. These allow for the most fine-grained inspections of subgroups of data. Both the ⓢ square selection and the ⓛ lasso selection tools provide the user with the ability to draw a box or a polygon around the points they desire to inspect further. Once the selection is made, the ③ *dimensionality reduction plot* changes to display the selection-polygon drawn by the user with a black dashed line, and highlights the selected points by lowering the opacity of the rest of the points. Furthermore, a subplot is appended to the top of the ② *cluster identification plot*, to display the time series in the selection in the same way as the clusters. Lastly, the ④ *selection summary display* also shows statistics of the points chosen. This graphic update is shown in Figure 3.11. Deselection is triggered by picking either of the selection tools and double clicking on the ③ *dimensionality reduction plot*.

4 Evaluation

This chapter lays out the experimental setup, and then reports on, and discusses the results of the studies and the survey.

4.1 Experimental Setup

To test the ClAIrify framework, the implementation described earlier underwent one experiment and two use case studies.

The experiment was designed to answer the research question as accurately as possible. The secondary research questions serve to answer the main research question, and thus, we had to settle the definitions of some words contained in them.

Our aim was to test the validity of our framework in relation to comparing machine learning performance, evaluating model performance, and understanding model performance. At the same time we wanted to prove that in fact it was more insightful than classical evaluation metrics to this end.

Thus, we went on to test ClAIrify in this light, based on the following definitions, already touched upon in chapter 1:

- *compare machine learning model performance*: estimate, measure or note the similarity or dissimilarity in performance of different machine learning models.
- *evaluate machine learning model performance*: form an idea/assess the quality of performance of machine learning models.
- *understand machine learning model performance*: perceive/understand the intended meaning of machine learning model performance in relation to the data.

As our framework was instantiated as a web-app, the approach to its validation was similar to a usability test. The experiment consisted of a Think Aloud Protocol (TAP), which is a standard format for testing usability, and which was adapted for our specific requirements.

As with any research in this academic institution, the Ethics and Privacy Quick Scan of the Utrecht University Research Institute of Information and Computing Sciences was conducted. It classified this research as low-risk with no fuller ethics review or privacy assessment required, which allowed us to carry it out.

Firstly, we had to select a specific profile of participants. In particular, we needed people that had a considerable knowledge of Artificial Intelligence (AI). To this end, we asked

3 second-year students of Utrecht University's Masters' in Artificial Intelligence to take part in it, as this assured us an adequate level of knowledge in the field. The study took place at the main researcher's residence (Oscar Alexander Kirschstein Schafer), on his personal computer. Before beginning the TAP, participants had to sign an information sheet and an informed consent form.

A TAP study consists in the participants fulfilling a series of exercises prompted by the researcher, while vocalising their thought process as much as possible. As we were testing complex concepts in view of the research questions, such as understanding, evaluation and comparison of models, we required the participants to have some base knowledge about the framework, so as not to compromise the study for lack of familiarity with the dashboard. Thus, a quick introduction was given to them, to be aware of the content of the *evaluation data* that was used, as well as an overview of the purpose of each plot.

During the whole duration of the series of exercises, the participant's voice and screen were recorded, using Windows 10's native Xbox Game Bar screen recording functionality and an external microphone plugged into a sound card. The participants were instructed to vocalise their thought process as much as possible, while trying to keep the instructor's intervention to a minimum. The instructor intervened a few times to give the participants some guidance when they strayed off the main activity, seeming to have forgotten the objective.

The activities the participants performed were designed to gradually increase in complexity, in order to first familiarise the participants with the environment, building up to end up rationalising across the entire dashboard. They consisted of the following three steps:

- **TASK 1**: Look at the model-cluster performance matrix and find which the best and worst performing models are.

- **TASK 2**: Select a cluster that is problematic and locate it both on the cluster graph, and the dimensionality reduction graph. With the knowledge about the data set, and looking at the cluster plot and stats, try to give a semantic meaning to the cluster.

- **TASK 3**: Now find the worst model-cluster pair, by looking at the matrix. With this information, filter the dimensionality reduction plot by selecting the according cluster and model. Inside this cluster, find any instances that are performing noticeably worse than the rest. Select them and try to give them a meaning, and compare them with the total instances of the cluster.

After the participants had completed all the tasks, they were given a questionnaire to fill out. Having in mind [5]'s study categorising usability questionnaires, we based ours on the Usability Metric for User Experience (UMUX), as presented in [21], as it offers post-study evaluation on satisfaction, effectiveness and efficiency. We adapted the questionnaire to ask about the three tasks relevant to our research questions, inquiring about model performance comparison, understanding, and evaluation. Lastly, we added one category to let the users evaluate their perception of the framework versus the classical evaluation metrics. Each of the four categories contains three questions, answerable through a Likert-scale of 7 points, which [57] suggest to be the most appropriate.

The survey was created using the Microsoft Forms platform, using the main researcher's Utrecht University student account, to comply with the university's ethical regulations.

4.2 Think Aloud Protocol

After finalising the study, we set out to evaluate its results. As is typical in TAP studies, the first step was to transcribe the audio of the recordings. The screen recordings already had the audio recording integrated, so we used these files to this end. We used OpenAI's Whisper tool [47], to make a baseline transcription. This tool was used because of its high accuracy and consistency, providing a close-to human-like performance [49]. We decided to use thematic analysis, as our study is close to a usability study, and this form of analysis has been used in similar studies, e.g. to analyse the usability of virtual health visits [23] or of a higher-education e-learning platform [1].

Running the Whisper tool on the MP4 recordings results in a few files representing different formats of transcriptions. We processed the resulting Javascript Object Notation (JSON) files with a custom Python script to generate a transcription file suited to be imported to NVivo, where the thematic analysis took place. NVivo 14 is a qualitative research software tool developed by QRS International. By actively reviewing a playthrough of the transcribed videos, we corrected any error made by the transcription tool.

Our thematic analysis is deductive, as we already had some preconceived themes beforehand, revolving about the tasks relevant to our research, i.e., (i) *understanding*, (ii) *evaluating* and (iii) *comparing* (machine learning performance) At the same time it is also inductive, because the specific codes in each theme were not decided beforehand, but came up while analysing the transcribed videos. Our approach is also semantic, as it only analyses the explicit content of the data, without taking into consideration the participants' context or speculating about the participants' assumptions. We follow the widely used six steps of thematic analysis proposed in [10]: (i) familiarisation, (ii) coding, (iii) generating themes, (iv) reviewing themes, (v) defining and naming themes and (vi) producing the report.

Familiarisation, i.e. understanding the transcription, was ensured by reading the transcript at least thrice, once for the correction, once for the coding part and once for proof-reading. Coding, i.e. highlighting sections of our transcriptions in relation to its meaning in regards of the study was then performed using the NVivo tool's functionality. In this stage, we came up with a fourth theme, in line with the partially inductive nature of our approac: *Design*. It was deemed as useful, as there were many comments made by the participants criticising the actual implementation of the framework, which is indirectly also related to the viability of the abstract, conceptual framework we set out to validate. After this first coding run, we ended up with the codes displayed in table 4.1. The codes we came up for each of the four themes are the same three:

- **Con** marks utterances that express displeasure with our framework's ability to fulfil the task described by the theme the code belongs to.
- **Confusion** marks utterances that express confusion about how to go about the task.
- **Pro** marks utterances that, opposite to *Con*, express a favourable disposition to use our framework for said tasks.

The coding was done taking into account three important aspects of the utterance.

1. The semantic content of the utterance.
2. The inherent theme/s of each task:
   - Task 1 - Comparing
   - Task 2 - Evaluating & Understanding
   - Task 3 - Comparing & Evaluating & Understanding
3. The actions the participant was performing in the app at the time of vocalising the utterance, as seen in the screen recording.

The third step of thematic analysis, *generating themes*, was completed by joining the three preconceived themes with the fourth theme that was suggested during the *coding* step. The fourth step, on the other hand, *reviewing themes*, serves to update, merge, split, expand or delete codes and themes in order to best validate the research. To this end, we played through all the videos together with their transcriptions. From here on out, each of the three participants will be referred to as P1, P2 and P3, while theme and code and code assignations will be referred to as **Theme.Code**.

In our case we decided to undo our addition of the *Design* theme, because, while it is useful for improving the concrete implementation, it doesn't serve our purpose of answering the research questions. Many of the utterances of this theme also had

Code	#part	#refs
Comparing		
Con	2	2
Confusion	1	2
Pro	3	24
Evaluating		
Con	1	1
Confusion	1	1
Pro	3	9
Understanding		
Con	2	1
Confusion	2	2
Pro	3	20
Design		
Con	1	3
Confusion	2	6
Pro	3	12

Table 4.1: Table indicating the distribution of codes the names of which are displayed in the leftmost cells of the white-colored rows. The central cells of these rows contain the number of participants that made utterances that this code applied to, while the rightmost cells count the individual utterances that this code applied to across all participants. The grey multi-column rows display the names of the overarching themes that each of the codes below belongs to.

another assignation to one of the initial three themes, which was kept. An example of this is:

- Utterance by P1: *"Is the one on the left because it's sorted, it's sorted according to performance, right?"*
- This utterance, referring to the ① *performance matrix plot*, was classified as **Design.Pro** because the design choice of sorting the columns with respect to average performance, seemed to prove meaningful to P1, but it was also labelled as **Comparing.Pro**, because the plot helped the participant to compare models with more insight.

Furthermore, we were able to reassign some of these utterances, previously only coded under *Design*, to another one of the themes based on the actions the user was performing in the screen recording:

- *Utterance* by P3: *"And here it's presented pretty well."*
- This utterance, referring to the overall design was classified as **Design.Pro**, due to it making reference to the design choice of placing the ① *performance matrix plot* at the top of the app being helpful. This was then relabelled to **Comparing.Pro** because of the utterance being in reference to the comparison plot being placed at the top, which potentially ensures a good entry point to analysis. Another argument for this is that this was said during Task 1, which is mainly focused on model comparison.

Apart from this, the code *Confusion* was also eliminated from all the themes. The affected utterances were instead reassigned to *Con*, as it was deemed that confusion is a negative aspect when either comparing, evaluating or understanding ML model performance in a framework that has the main purpose of clarifying. The resulting code distribution, this time shown also per participant, is shown in Table 4.2. With respect to the previous coding version, the difference in codes is displayed in Table 4.3, where it is shown that there was an overall growth in codes for every coding category.

Code	P1	P2	P3	#refs
Comparing				26
Con	0	0	5	5
Pro	8	6	7	21
Evaluating				14
Con	2	0	1	3
Pro	1	4	6	11
Understanding				35
Con	1	1	2	4
Pro	11	11	9	31

Table 4.2: Table indicating the distribution of codes the reviewed code count. It shows the total count for each code, and theme, as well as how each code is distributed across each of the three participants.

The last two steps, *defining & naming the themes* and *writing the report* were trivial, and the former had already been done.

In general, we found that P1 and P3 were more prone to stray off the activities, which could be due to a lack of concentration, or a lack in experience of performing a study of this sort. Another reason may be that the design of the tasks and information put a high cognitive load on the participants, affecting their concentration/memory. This could bias the results, which was one of the reasons why the instructor provided guidance when the participant seemed to be lost. Example:

Code	Δrefs
Comparing	6
Con	4
Pro	2
Evaluating	6
Con	2
Pro	4
Understanding	10
Con	5
Pro	5

Table 4.3: Indicates the difference in coding between the first and second coding procedures.

1. P1 starts to forget the aim in completing Task 1, which is to identify the best and worst overall performing models. They only answer partially (to the question: which is the worst model?): *"So this is like Long-Short Term Memory (LSTM) is the worst performing model."*
2. Instructor notices they don't answer to the other part of the question, and prompts: "And which is the best?"

Another aspect noted during the study was that P3, at certain times, for example, when trying to find the worst and best performing clusters (in terms of performance score, i.e. Root Mean Square Error (RMSE)), instead evaluated the clustering quality in terms of how compact the clusters appeared in the dimensionality reduction plot. In the end, they were guided to complete the tasks correctly.

The times of completion for the tasks of each participant are represented in Table 4.4. The average total completion time was ∼12:20 minutes. For Task 1, the average time was 02:57 minutes, for Task 2, 04:21 minutes and for Task 3, 05:01 minutes. This increase in completion time was expected, as the tasks were increasingly more complex.

Participant	Task 1	Task 2	Task 3	Total
P1	03:35	03:07	4:41	11:26
P2	00:38	02:18	04:10	07:06
P3	04:38	07:40	06:12	18:30

Table 4.4: Shows the times the participants took to complete each task, and how much it took them in total. The format is *minutes:seconds*

4.2.1 Comparing

In relation to ML model comparison, only P3 made negative statements (**Comparing.Con**). They firstly mention that they deem the numbers inside each cell of the ① *model performance matrix* to be quite arbitrary, even though they specify it may only be so for a layman user:

- P3 in Task 1: *"But like these numbers inside of it are just like quite arbitrary. But if I look here to the right, I can see root mean square error. And, you know, because I am in this field, I do understand what that means. But like, I just feel like somebody could look at this and be like, okay."*

This hints towards the need of specifying further cues towards the meaning of these. Furthermore, P3 criticised the colour scheme used as being non-inclusive:

- P3 in Task 1: *"And also, for colourblind people, that may be a problem because this screen and this screen could basically be the same colour for them."*

This is an important aspect to take into account for an interface that, even though it is meant to be used by specialists, it is still meant to be shown to everybody, so it may mean that this should be an important part to include in the guidelines.

Both of these negative aspects relate to the implementation, and were not specified in the guidelines.

As far as the positive comments (**Comparing.Pro)**, they often make reference to the easiness to complete certain tasks. For example, P1 states the obviousness of where to look at to compare models:

- P1 in Task 1: "*Obviously up here. So LSTM is the worst model. So this, yeah this gives you the average across the different clusters, right?*"

Also, for finding the best performing model, they state:

- P1 in Task 1: "*Its the one on the left because it's sorted, it's sorted according to performance, right?*"

The first statement refers to the ① *performance matrix plot* being obviously the tool to compare overall model performance, while the second one talks about the design choice made in the implementation of sorting the columns of the matrix according to performance.

P2 made a comment relating to the intuition behind using red as a colour that helps identify problematic instances:

- P2 in Task 1: "*I should look for red because red is problematic. So then I'll look for the highest, which are cluster one and cluster three.*"

In Task 3, P3 also quickly points out the correct answer:

- P2 in Task 3: "*...cluster one with the LSTM is the worst cluster model pair.*"

P3 talks about the diversity of options to go about comparing performance:

- P3 in Task 1: *"...so there is a couple of ways I could go into that..."*

This potentially makes reference to the fact that, even though the ① *performance matrix plot* is the main informative vessel for model comparison, it would also be possible to compare performance looking at the hue of the data points in the ③ *dimensionality reduction plot*. to get an intuition about overall performance.

Furthermore, they mention the positive utility of indicating the units on the colour-bar of the ① *performance matrix plot*:

- P3 in Task 1: *"...like how you did it here for to the right, these bars where they say it's Fahrenheit. Because somebody may not know what root mean square it is, but you can recognise the Fahrenheit and make the assumption, okay, I can say that green could be cold, but red is definitely very warm. So you instantly like understand what the thing that's being displayed here. "*

In general, the positive comments in relation to comparison potentially indicate that the framework enables extremely easy model performance comparison.

4.2.2 Evaluating

P1 in the process of trying to evaluate which the worst performing cluster is, looks at the ③ *dimensionality reduction plot* and makes a negative comment (**Evaluating.Con**):

- P1 in Task 2: *"So the thing is like the dimensionality reduction, like I can see the clusters like separate from each other, but I don't really understand like what are the axes."*

It seems they were confused about the utility of the dimensionality reduction plot, and while the axes' units are unknown due to the nature of dimensionality reduction, closeness in this dimensional space should give intuition about similarity between instances. This should make it a matter of *Understanding*, not *Evaluating*. Still, the fact that this conceptual error came up when trying to evaluate performance, speaks to a potential necessity to make the dimensionality reduction's advantages more obvious.

P3, as mentioned before, strays off Task 2, starting to evaluate the clustering quality when trying to find the worst performing cluster:

- P3 in Task 2: *"This is cluster three, just because it has like a massive gap where values are separated like on two other, like different extremes."*

While this may be seen as a conceptual error of the participant, which could have many causes (e.g. lack of familiarity with the app, lack of preparatory information in the study...) it potentially also speaks to the same problem addressed by the previous participant.

As for positive evaluation (**Evaluating.Pro**), P1 points out that they recognise that the worst performing model-cluster pair is LSTM on the first cluster:

- P1 in Task 3: *"So yeah, it will be probably the LSTM with the first cluster."*

P2, again, quickly points out the helpful cues that the colour scheme gives towards evaluating, in regards to finding the worst performing model-cluster pair:

- P2 in Task 3: *"Noticeably worse is red...So this one is quite worse."*

P3, makes the following statement in Task 3, after filtering the dimensionality reduction plot with the worst performing model of the worst performing model-cluster pair:

- P3 in Task 3: *"And I feel like these, okay, these scores are really bad...The ones at the bottom are pretty good. "*

P3 can point out the points that perform better or worse thanks to the colour scheme. So even though, earlier this was criticised by them for lack of inclusiveness for colour-blind people, it still results useful.

These positive comments on evaluation make more of an emphasis on the evaluation going on in the ③ *dimensionality reduction plot*, hinting at it being potentially useful for such use cases, complementing the ① *performance matrix plot*.

4.2.3 Understanding

P1 again makes a statement regarding dimensionality reduction that is deemed as negative (**Understanding.Con**):

- P1 in Task 2: *"Okay, let's try a few different, a few different dimensionality reduction methods just to give us an idea of what's going on."*

This comment is made in regards to giving a semantic meaning to the cluster. It is deemed negative because different dimensionality reduction techniques don't offer a different semantic understanding of a cluster. What is intended to do is to look at the ② *cluster identification plot*. P1 seems to fail to acknowledge the use of the dimensionality reduction. It may be potentially because the participant is not sure of what to do in general.

P2 makes a comment in Task 3 about the limitations of the ③ *dimensionality reduction plot*:

- P2 in Task 3: *"So even when I will select them, which I'm going to do now. I will select some points that are quite all right."*

This utterance talks about the difficulty to select only the points that want to be evaluated. In this specific case, there are some points that offer a better performance in between the bad performing points. This is a problem inherent in the dimensionality reduction plot.

P3 points out his lack of knowledge of the domain that the data belongs to:

- P3 in Task 3: *"Yeah, I just I can I can see but I'm not quite sure what is causing them to be this bad because, you know, I'm not doing climate stuff."*

This points out one of the requirements of the framework, i.e. that the user of ClAIrify must already be considerably familiar with the data they are working with.

As for the positive comments in this area (**Understanding.Pro**, P1 states:

- P2 in Task 2: *"And I guess that the more drastic the temperature change is, the more difficult it is to predict, for the models to predict the temperature."*

This is an attempt at inferring the reason behind a specific cluster being more difficult to be predicted on. In fact, it may well be the case that, due to using vanilla models, that barely take into account the time series nature of the data, the instances with more protuberant oscillations (*"...the more drastic the temperature change is..."*) are more difficult to model.

P2, in Task 3, finds semantic meaning in the problematic cluster, saying:

- P2 in Task 3: *"Very hot summers is what you see here."*

This is said after looking at the ② *cluster identification plot*, which is exactly what is intended.

Finally, another interesting utterance is made by participant 3, who, while completing Task 2 says:

- P3 in Task 2: *"And that could mean that they just, they just have the same temperature or like are similarly tropical. And if I look at cluster one and cluster five, these two graphs do correlate quite a bit. So like that could make an assumption that the climates in those two, two different cities are very, very similar."*

This points out one of the potential strengths of the ③ *dimensionality reduction plot*, making the participant realise that two instances of atmospheric temperature time series belonging to cities being close in the dimensionality reduction plot may indeed mean that these two cities are close geographically speaking.

4.3 Survey Results

This section reports on the results of the post-TAP survey explained in section 4.1. The survey's purpose was to quantify the qualitative opinion of the participants with the intention of answering the RQs.

Figure 4.1 shows the results of the answers to the questions regarding ML model performance comparison. It can be appreciated that answers to the questions give the intuition that using the ClAIrify framework for model performance comparison potentially fulfilled the participants requirements. Moreover, it was not frustrating, but rather easy. Strongest answers were stated towards saying that ClAIrify is easy to use for comparison (2/3 of the answers being *strongly agree*), and for denying frustration, with 1/3 of the answers being *strongly disagree* and 2/3 being *disagree*.

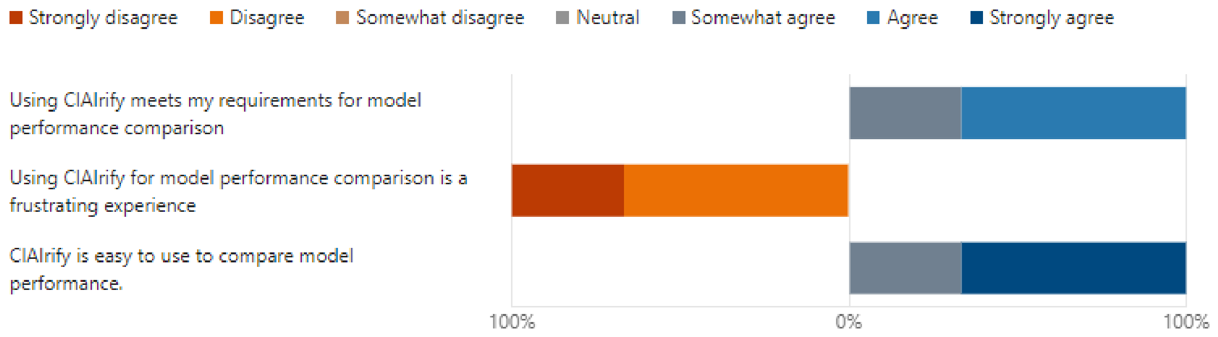


Figure 4.1: Likert answer distribution for ML model performance comparison questions.

Figure 4.2 shows the results for questions related to ML model performance evaluation. The distribution for the answers is rather similar to the ones relating to comparison, potentially hinting at the viability of the framework for use to this end.

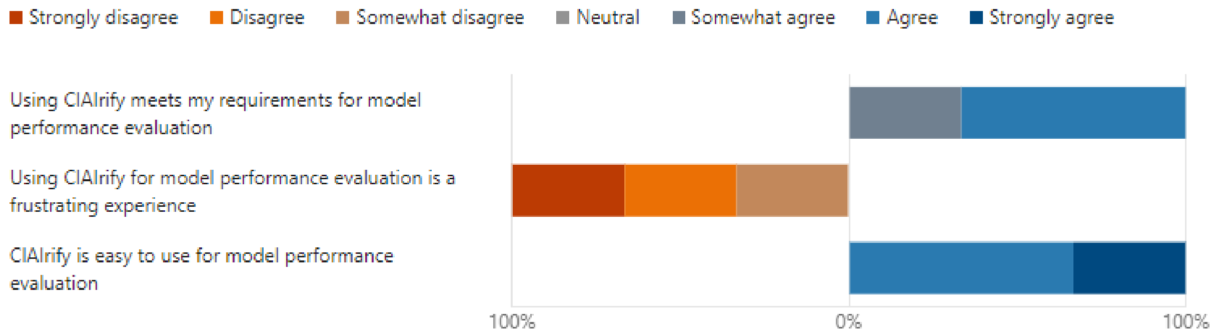


Figure 4.2: Likert answer distribution for ML model performance evaluation questions.

The answers for the likert questions relating to ML model performance understanding are displayed in Figure 4.3. Here we can see more moderate answers in general. The framework's easiness to use is considered neutral (2/3), except for one answer which agrees strongly. One participant disagrees somewhat that the framework helps for understanding. In terms of frustration, while 2/3 disagree, 1/3 has a neutral opinion with respect to it.

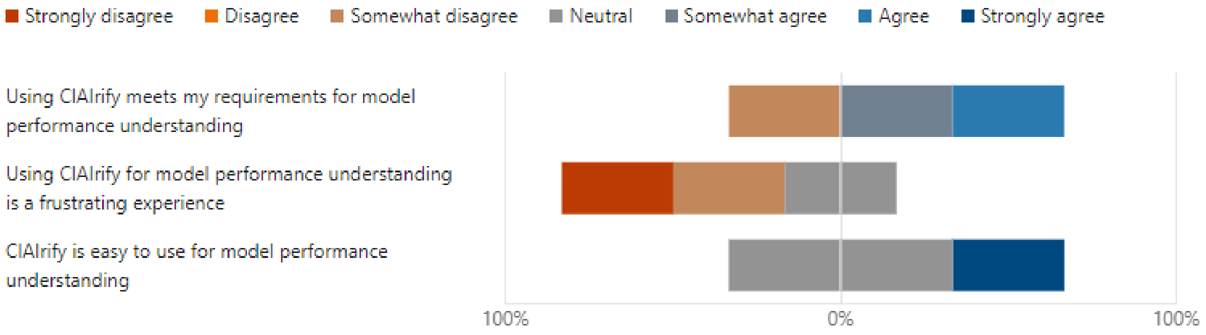


Figure 4.3: Likert answer distribution for ML model performance understanding questions.

When it comes to comparing the framework with other classical metrics (Figure 4.4), such as Mean Average Performance Error (MAPE) or RMSE, the participants had a rather positive feedback in general. 1/3 strongly agreed and 2/3 agreed that they would prefer using ClAIrify than the traditional metrics for performance comparison. They also generally preferred ClAIrify for evaluation (1/3 neutral and 2/3 *Agree*). Still, one participant answered *Somewhat disagree* to preference of ClAIrify over the traditional metrics for performance understanding. Another one answered *Neutral* to this question, and another one *Agree*.

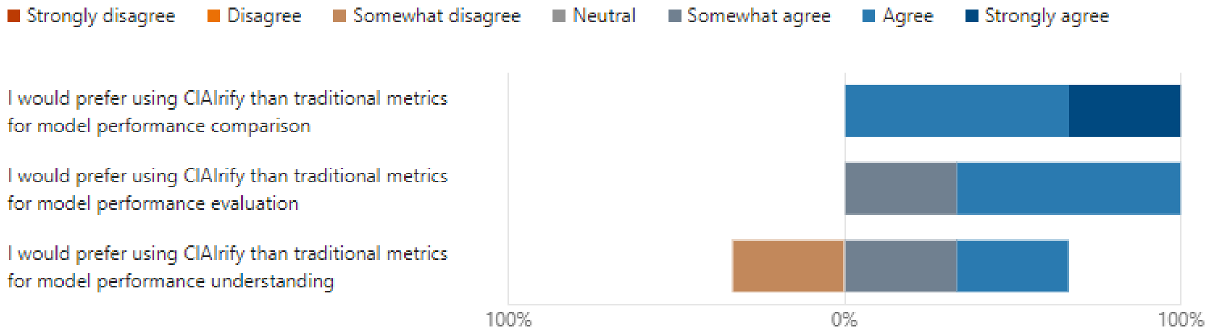


Figure 4.4: Likert answer distribution for questions regarding classical metrics compared with our framework implementation.

There was also a section for free feedback answers, to gather further insights from the participants. Only two of the participants opted to answer this. The answers were the following:

- *"The dimensionality reduction plot is not intuitive without knowing more about the dimensionality of the data to begin with"*
- *"I found it not really frustrating but it could be an improvement if you could select multiple points and not a whole lasso or rectangle."*

One participant indeed believes that the ③ *dimensionality reduction plot* should be given some more background information in order to become more useful, something already hinted at when looking in the coding analysis section.

Furthermore, the other participant believes it to be useful to extend the interactivity of the ③ *dimensionality reduction plot* by selecting multiple points (individually). This way, it would be easier not to accidentally select instances that are not deemed necessary for the analysis.

4.4 Use Case Studies

To further validate our framework, we have conducted two use case studies, on two different data sets. The studies empirically evaluate the web app implementation of ClAIrify described in chapter 3. Both of these studies use multiple uni-variate time series as a data set, and tackle a regression problem. They consist of monthly averages of weather and financial opening stock price values, respectively. The intention is to predict five steps into the future.

We prepared the data for the application to work, as necessary. First, we trained an array of models on the training data which consist of the time series data except the last five time steps in each sample. These last steps make up the test set. The models we trained are the following:

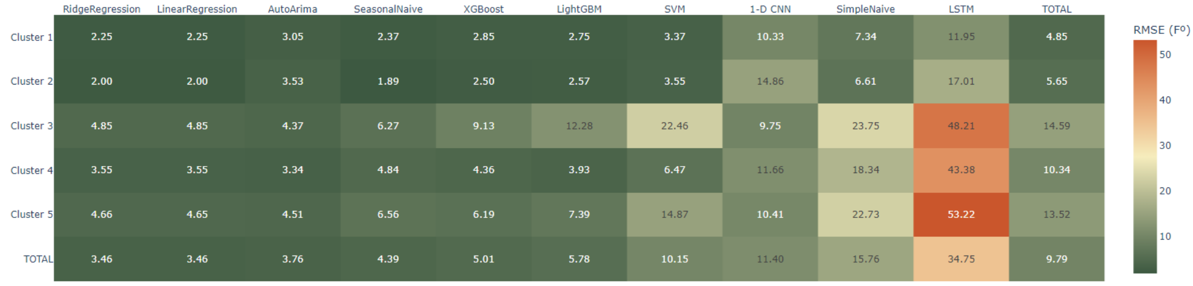
- Naïve forecast: A model that uses the last available training time step as the prediction.
- Seasonal Naïve: Predicts the value of the same month of last year.
- AutoARIMA: A version of Autoregressive Integrated Moving Average (ARIMA) that finds the required parameters automatically. These are fitted individually per series.
- Linear Regression.
- Ridge Regression.
- LGBM: Light Gradient Boosting Machine
- XGBoost: Extreme Gradient boosting.
- LSTM: Long-Short Term Mermory network.

- SVM: Support Vector Machine
- MLPRegressor: Multilayer Perceptron Regressor
- 1-D CNN: One dimensional convolutional neural network.

Where required, the models have been set up with the default parameterisation of their Python implementations. Once the models have been trained, we calculated the instance-wise evaluation metric. It was decided to use the RMSE metric, for its widespread use in regression tasks. Another advantage is that it is in the same units as the target variable, which allows for an enhanced interpretability. As requested, each time series has one RMSE value associated per model trained, that was calculated on the 5 predicted time steps compared to the 5 target time step values. For the analysis it is important to note that the lower the RMSE value, the better the result.

4.4.1 1st Use Case Study - Weather Data

This is the same data set that was used in the TAP. The data used is of the daily (but aggregated per month by us) temperatures, in Fahrenheit (°F) of 323 international cities, between January 1999 and July 2001. This dataset was shared by the Dayton University, Ohio[61].



	RidgeRegression	LinearRegression	AutoArima	SeasonalNaive	XGBoost	LightGBM	SVM	1-D CNN	SimpleNaive	LSTM	TOTAL
Cluster 1	2.25	2.25	3.05	2.37	2.85	2.75	3.37	10.33	7.34	11.95	4.85
Cluster 2	2.00	2.00	3.53	1.89	2.50	2.57	3.55	14.86	6.61	17.01	5.65
Cluster 3	4.85	4.85	4.37	6.27	9.13	12.28	22.46	9.75	23.75	48.21	14.59
Cluster 4	3.55	3.55	3.34	4.84	4.36	3.93	6.47	11.66	18.34	43.38	10.34
Cluster 5	4.66	4.65	4.51	6.56	6.19	7.39	14.87	10.41	22.73	53.22	13.52
TOTAL	3.46	3.46	3.76	4.39	5.01	5.78	10.15	11.40	15.76	34.75	9.79

Figure 4.5: Performance matrix plot showing performance of models on clusters of weather time series data of use case study 1.

By looking at the *model performance matrix*, displayed in 4.5, we can immediately make comparisons between performance of models and clusters alike and see that the cells displayed in a more intense shade of orange are the worst performing, as they hold higher values of RMSE.

Ridge and Linear Regression models both boast the best overall performance, being easy to find, as they are the leftmost models, with an RMSE of 3'46 °F. They are thus potential candidates to be chosen for use in production. The worst performing model is the LSTM model, with an error score of 34'75 °F. An error of this magnitude should rule out the usage of this model to make accurate predictions.

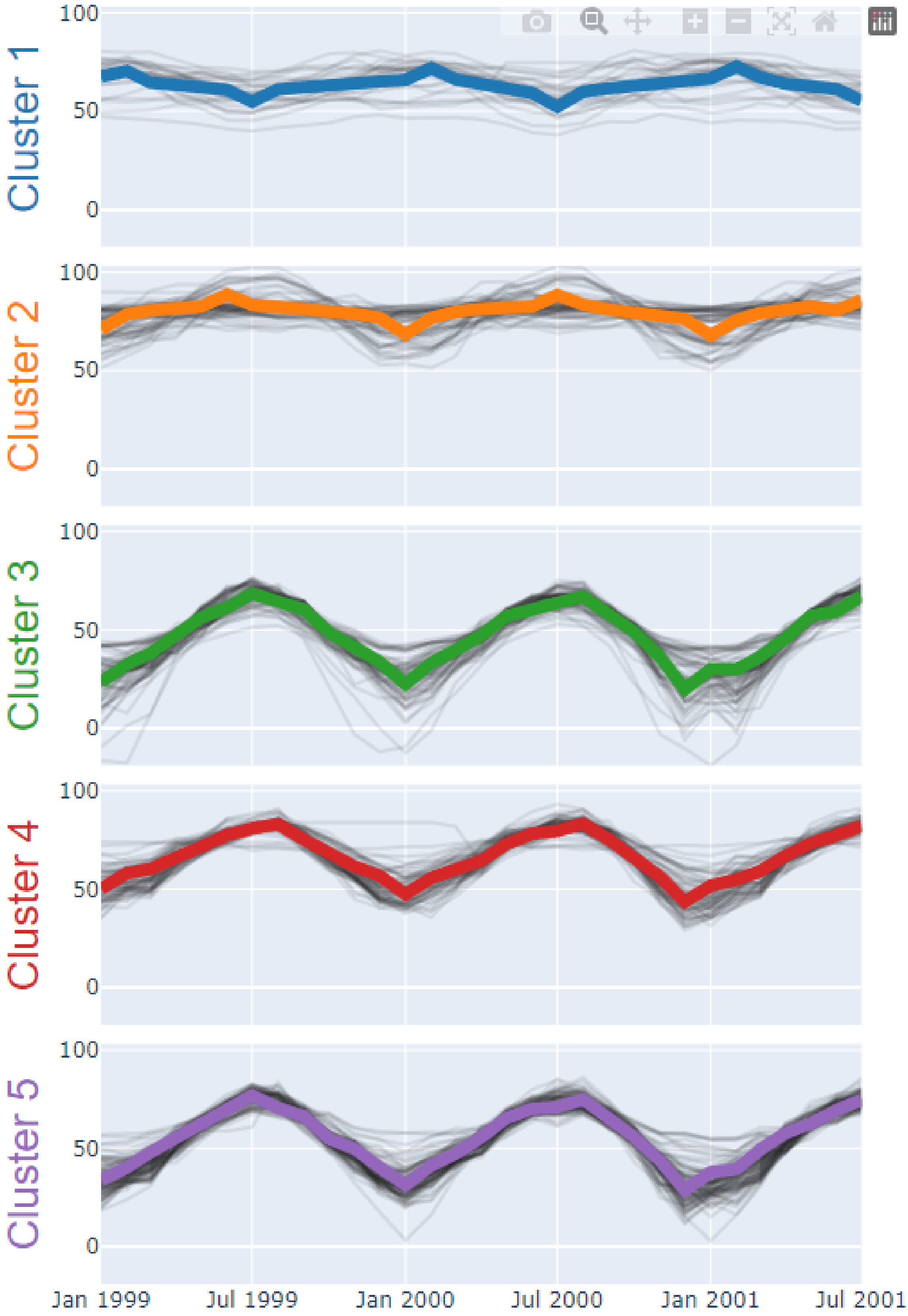


Figure 4.6: Cluster identification plot showing weather time series clusters of use case study 1.

It is important to interpret the semantic meaning of the clusters in order to gather an intuition about the semantic meaning of them, for which one ought to look at the *cluster identification plot* (Figure 4.6). The potential meanings we found are the following:

- **Cluster 1**: Shows a relatively flat time series (μ = 63.6476 °F (17.582 °C), σ = 8.5660 °F (4.7589 °C)). Behaviour like this is usually a sign of places close to the equator. Furthermore, it can be observed that the lowest values concentrate around July and the highest around February. These most likely correspond to the winter and summer months, respectively. Such a weather pattern is typical for countries in the southern hemisphere.
- **Cluster 2**: Also shows a relatively flat average behaviour (μ = 79.1412 °F (26.1896 °C), σ = 7.6084 °F (4.2269 °C)), though higher on average. The difference between this cluster and **Cluster 1** is that the average temperature is higher, possibly implying that the cities this data belongs to are even closer to the equator. Moreover, the summer and winter months are switched with respect to the previous cluster (June and January, respectively), indicating that the data belongs to countries in the northern hemisphere.
- **Cluster 3**: It immediately stands out that this cluster displays a much higher variance of temperature over the months than the previous clusters (μ = 45.9757 °F (7.7643 °C), σ = 16.7861 °F (9.3256 °C)). The summers around June and winters around January imply possible location in the northern hemisphere, with the high variance indicating that the location is more likely far away from the equator than previous clusters.
- **Cluster 4**: Also shows a high variance (μ = 65.9327 °F (18.8515 °C), σ = 13.2642 °F (7.3690 °C)), though slightly less than **Cluster 3**, while also displaying a higher mean temperature. This, together with the summer-winter pattern possibly identifies locations in the northern hemisphere that are closer to the equator than the cities in the previous cluster.
- **Cluster 5**: Taking into account this cluster's mean and standard deviation (μ = 54.1767 °F (12.3204 °C), σ = 16.0184 °F (8.8991 °C)), and following the logic used for the previous clusters, it seems that the locations corresponding to this data are situated in the northern hemisphere, possibly closer to the equator than the cities in **Cluster 3**, but further away than the ones in **Cluster 4**

What becomes apparent when combining the information from the *performance matrix plot* and the *cluster identification plot* is that the worst performing clusters are the ones that display most variance in temperature, i.e. clusters 3, 4 and 5. This conclusion was also reached by participants in the TAP study, which implies that potentially there can be a common understanding generated through the framework.

Figure 4.7 shows the *dimensionality reduction plot* for this case study, with **Cluster 3** selected (shown in the highlighted green polygon) with the ⓑ *cluster selection dropdown*.

This cluster was selected because it had the worst overall performance, but also more specifically for the two best models: *Ridge Regression* and *Linear Regression*.

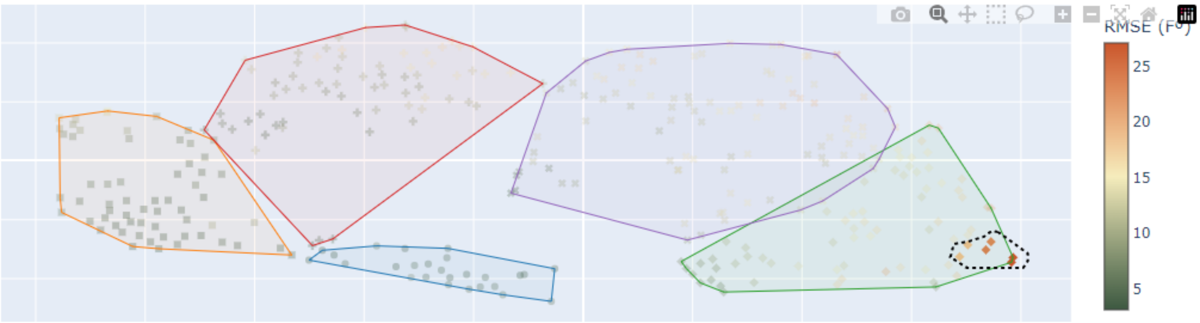


Figure 4.7: Dimensionality reduction plot showing performance of models on clusters of weather time series data of Case Study 1. Bad performing instances have been selected with the lasso selection tool.

The best performing data instances can be seen more towards the left, while the ones on the right are worse, which can be evaluated easily with the green-red hues. Here it may potentially be valuable to look into the data points to see what issues may be causing a loss in performance, and thus some instances on the right-lower corner, as seen in the Figure 4.7, have been selected.

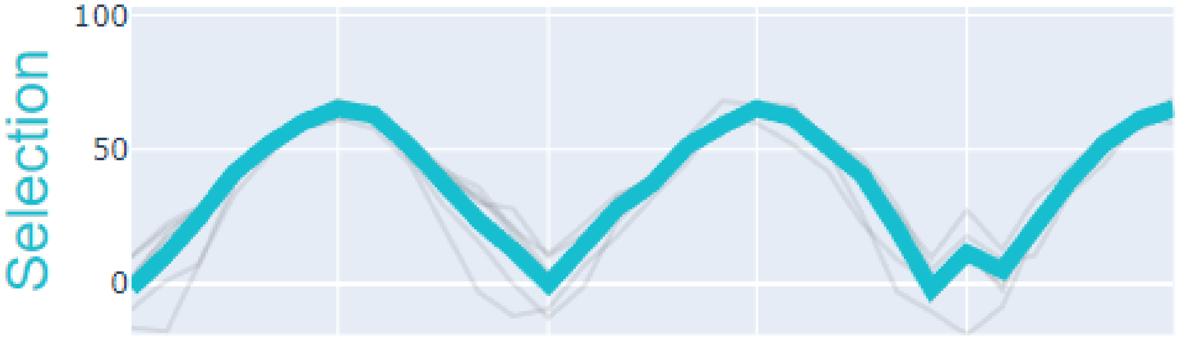


Figure 4.8: Subplot of the *cluster identification plot* showing time series present in selection made in the *dimensionality reduction plot*. It represents the worse performing data points of Cluster 3 in Case Study 1.

To gather further insight into the selection, the instances are plotted in a subplot at the top of the ② *cluster identification plot*, and looks as shown in Figure 4.8. Comparing the mean μ = 35.6939 °F (2.0521 °C) and standard deviation σ = 8.5468 °F (4.7482 °C)) of the selection to the the ones of **Cluster 3** (μ = 45.9757 °F (7.7643 °C), σ = 42.5534 °F (23.6408 °C)) , it becomes apparent that the time series from the selection have more than double the variance (2.7661), which points to a pattern that has been apparent during this case study, and also found by the participants in the TAP study. The worse performing instances belong to the time series of cities with a very high variance in atmospheric temperature throughout the year. Thus, it becomes clear that something must be done so that this variance does not affect prediction performance. Something that may come to mind as a solution to this may be to add information about the month that is going to be predicted, as well as information about the latitude and longitude of the cities, as this was something that could also be extracted as a potential cause from analysis of the clusters.

This point is reinforced by seeing what happens when we select the better performing instances (Figure 4.9). The shape of these time series, as shown in Figure 4.10, portrays time series with a considerably lower variance (μ = 50.8609 °F (10.47827778 °C), σ = 9.9882 °F (5.549 °C)) than the ones of the cluster, which may indicate that indeed variance in the time series plays a big factor in performance. This portrays a prime example of potentially understanding the data and the cause for behaviour of the ML models.

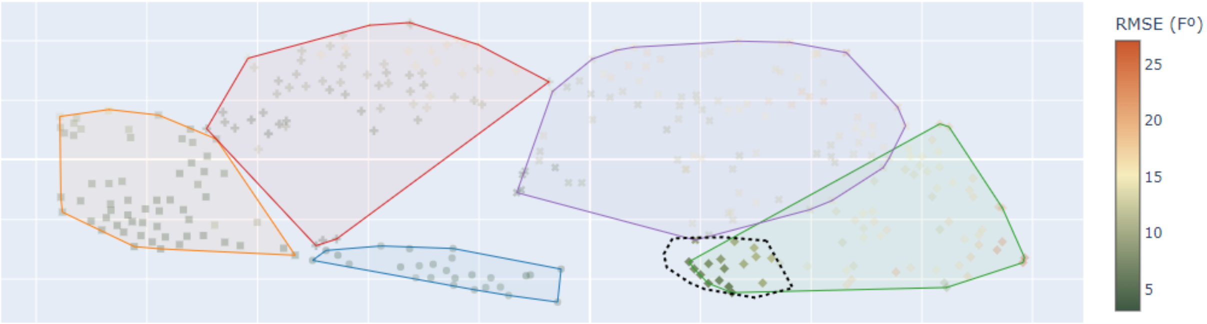


Figure 4.9: *Dimensionality reduction plot* showing performance of models on clusters of weather time series data of Case Study 1. Good performing instances have been selected with the lasso selection tool.

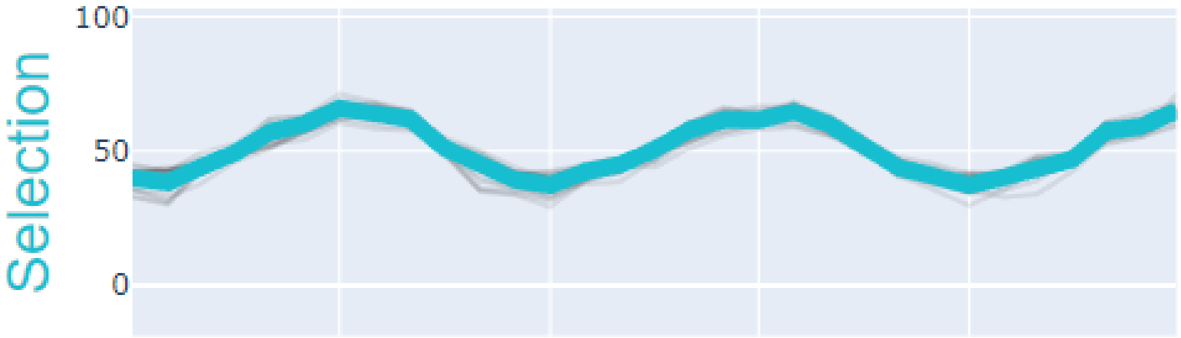


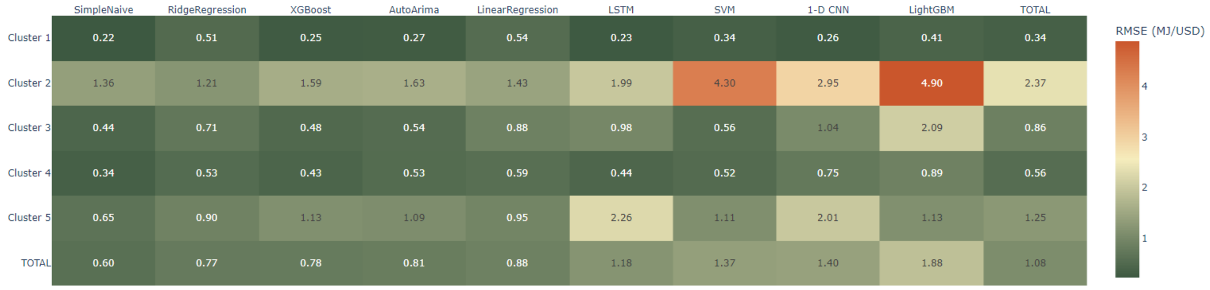
Figure 4.10: Subplot of the *cluster identification plot* showing time series present in selection made in the *dimensionality reduction plot*. It represents the better performing data points of Cluster 3 in Case Study 1.

4.4.2 2nd Use Case Study - Value Added Data

This study was also performed on a univariate time series data set. In this case, the data of the Energy and Emissions per Value Added Database [3]. Among other records, it contains the "per value added energy intensity" (Per-Value Added Energy Intensity (VAEI)) indicator for many economical sectors (i.e. *Construction*, *Mining and quarrying* & *Agriculture, forestry and fishing*). In our specific use case we are interested in the evolution of the yearly VAEI indicator for the *Agriculture, forestry and fishing* sector of each country. It is measured in 'Mega joules per US Dollar' (MJ/USD) and shows the amount of energy used per unit of economic output, in this case, in the aforementioned sector. It measures the efficiency of an economy, as lower VAEI value indicates lower cost of converting energy into Gross Domestic Product (GDP).

The data consists of time series of 64 countries representing the yearly VAEI index, over a period of 22 years, from 2000 to 2021. The aim was to train models to make a long-term predicition of the next 5 years of VAEI, for each country. Thus, the last 5 years were reserved for the test set, and the models were trained on the remainder of the time series.

The overall performance of the models can be observed in Figure 4.11. It is apparent that the models perform differently from case study 1. Moreover, Ridge and Linear Regression don't perform the same anymore, and this time the Simple Naive approach has become the best performing model, while the LightGBM model is the worst. Again, with help of the colour scheme, it is easy to identify that the worst overall performing cluster is **Cluster 2**, with an average RMSE of 2.37 MJ/USD.



	SimpleNaive	RidgeRegression	XGBoost	AutoArima	LinearRegression	LSTM	SVM	1-D CNN	LightGBM	TOTAL
Cluster 1	0.22	0.51	0.25	0.27	0.54	0.23	0.34	0.26	0.41	0.34
Cluster 2	1.36	1.21	1.59	1.63	1.43	1.99	4.30	2.95	4.90	2.37
Cluster 3	0.44	0.71	0.48	0.54	0.88	0.98	0.56	1.04	2.09	0.86
Cluster 4	0.34	0.53	0.43	0.53	0.59	0.44	0.52	0.75	0.89	0.56
Cluster 5	0.65	0.90	1.13	1.09	0.95	2.26	1.11	2.01	1.13	1.25
TOTAL	0.60	0.77	0.78	0.81	0.88	1.18	1.37	1.40	1.88	1.08

Figure 4.11: Performance matrix plot showing performance of models on clusters of VAEI time series data of use case study 2.

Looking at Figure 4.12, a possible interpretation of the semantic meaning of the clusters that were generated is as follows:

- **Cluster 1**: This cluster contains the VAEI evolution of the countries with the most energy efficient $\mu = 0.512$, $\sigma = 0.5037$ *Agriculture, forestry and fishing* sector.

- **Cluster 2**: The data inside this cluster, is the opposite of the previous one, containing the data of the least energy efficient countries $\mu = 12.5334$, $\sigma = 1.7514$. Still, there is a trend of improving their energy efficiency.
- **Cluster 3**: The data instances in this cluster are the third most energy efficient ones $\mu = 4.6682$, $\sigma = 0.948$. As with **Cluster 2**, there is an apparent improvement over time of this energy efficiency, though slightly less noticeable.
- **Cluster 4**: This cluster's data $\mu = 2.2936$, $\sigma = 0.9246$ possibly describes countries with a very high energy efficiency, being the second most energy efficient ones, between all the clusters. The time series remain constant throughout the years.
- **Cluster 5**: The data in this cluster belongs to the countries with the second least energy efficient $\mu = 7.0301$, $\sigma = 1.8088$ *Agriculture, forestry and fishing* sector. As clusters 2 and 3, it exhibits an improvement in efficiency over time.

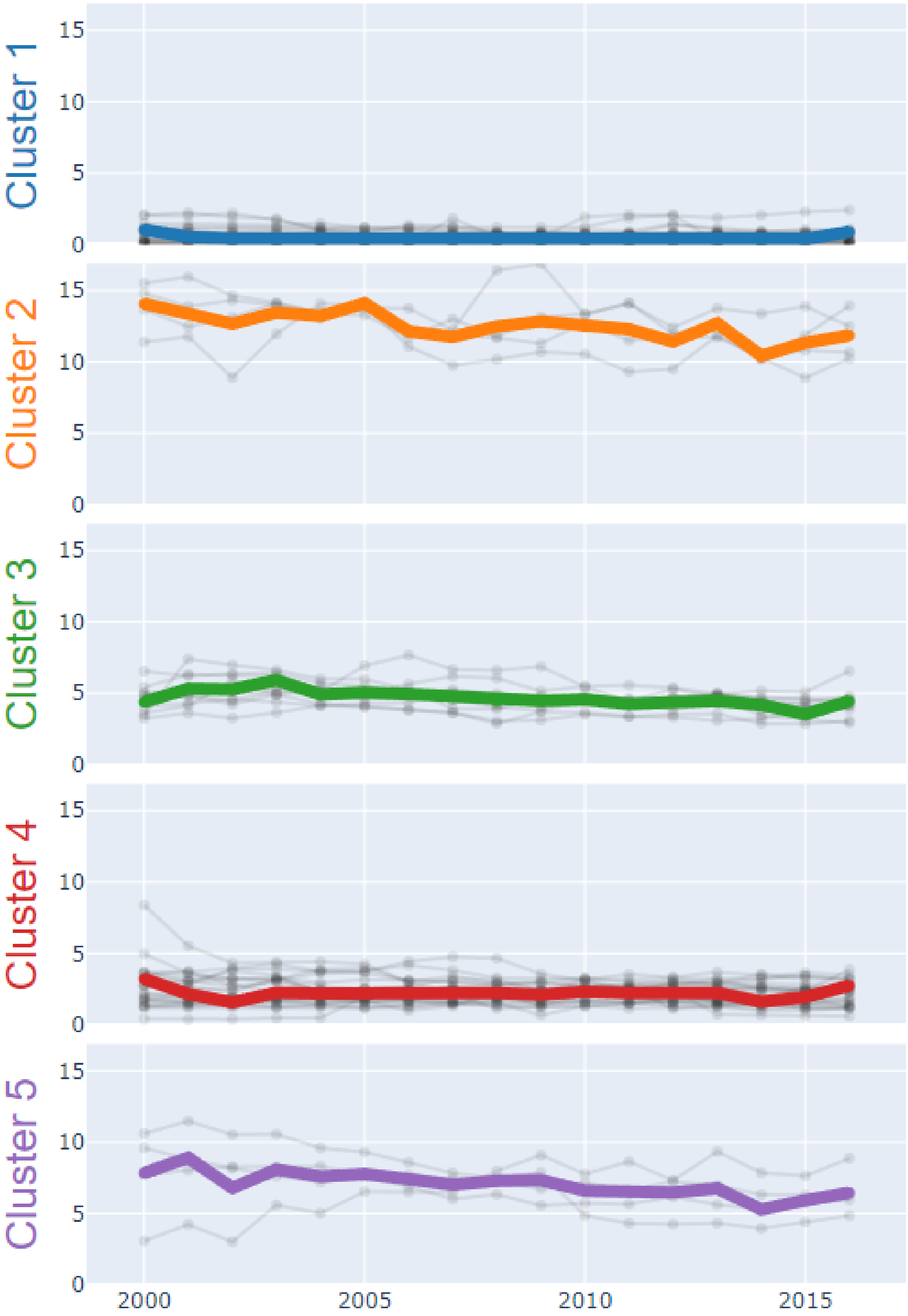


Figure 4.12: Cluster identification plot showing the VAEI time series clusters of Use case study 2.

It is valuable to look into Cluster 2, as it is the worst performing cluster, to potentially gain insight into what may cause this. When looking at Figure 4.13, and having in mind the previous cluster analysis, we can observe that it seems that the further to the right the data instances are, the least energy efficient the countries they belong to. Furthermore, it seems that, with some exceptions, the average performance of the models on each data point also gets better the more energy efficient they are. This makes it so that Cluster 2, our object of analysis, is the rightmost cluster. It contains only 4 data instances, which, after inspection, correspond to very irregular time series. As with Case Study 1, it is this irregularity that most likely causes the models not to fit the data accordingly. A possible solution to this poor performance is most likely to add additional data features to the input, such as other economical indicators, i.e. GDP.

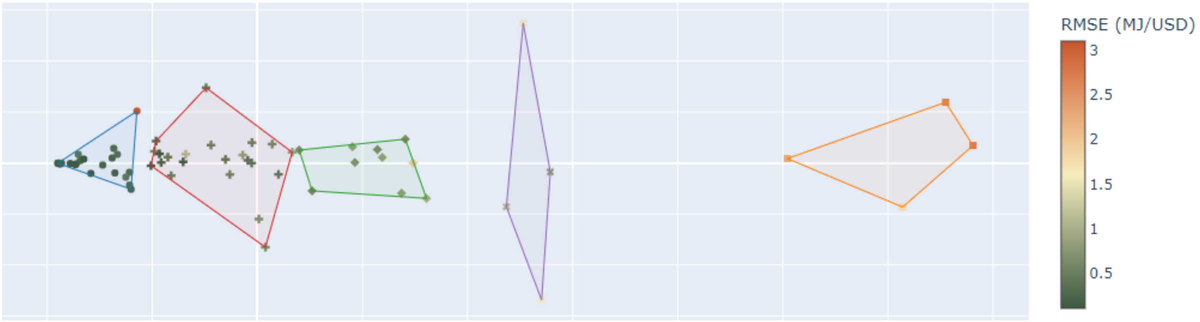


Figure 4.13: Dimensionality reduction plot showing performance of models on clusters of VAEI data of Case Study 2.

4.5 Discussion Summary

Merging the interpretation of the results from the TAP study, the survey and the case studies, we can attempt to answer the research questions. Firstly, it is useful to look at the first three secondary research questions, in order to be able to answer the main research question. **SRQ1**, **SRQ2** and **SRQ3** ask about the possibility to use our framework for comparing, evaluating and understanding machine learning performance, respectively. The results from all three of the evaluation methods point towards a positive answer. The transcription statements were coded with a considerably higher amount as positive (Pro) codes than negative (Con):

- Comparing - 21 Pro and 5 Con
- Evaluating - 11 Pro and 3 Con
- Understanding - 31 Pro and 4 Con

The contents of the statements, as seen before, reinforces this. The survey answers also hint towards this, as well as the results of the case studies. All the latter points towards the answer to the main research question, *"Can we use a data type-, task- and model-agnostic, clustering-based framework to compare, evaluate and understand model performance in a machine learning problem?"* being a yes, potentially. Still, we believe that it is necessary to do further research to test this, while also addressing the limitations, which are commented on in the next section. The evidence to answer **SRQ4** (*"Does a data type-, task- and model-agnostic, clustering-based framework give more insight when comparing, evaluating and understanding model performance in a machine learning problem than classical evaluation metrics?"*), relies solely on the answers to the survey. Here, all the participants agreed to some degree to the proposition of the research question, except for understanding, where one participant answered *Somewhat disagree* to them preferring ClAIrify over traditional metrics for model performance understanding.

5 Limitations & Future work

It is unavoidable that intending to answer research questions with such a subjective nature, and developing a conceptual framework with such a broad scope comes with its limitations, which will be discussed in this Chapter.

One important insight mentioned in [16] is that developers of a VA system, have to be careful not to create black box components in order to explain another black box, as that harms explainability. This is closely related to ClAIrify, because we certainly use some black box components to enhance explainability of machine learning performance, i.e. clustering and dimensionality reduction. The results of these algorithms can not always be explained. Furthermore, each run of the algorithms can give a different result, so they are not deterministic. Something that could be done as future work is to try to implement more explainable versions of these algorithms, and to require this explainability also in the guidelines. There has been some work done on this matter [14, 22], which could give a direction to look into.

Another shortcoming of this project is an assumption about the data. The implementation of our framework assumes that the clusters to be found in the data are convex. The k-means clustering algorithm, the silhouette score and the plotting of convex hulls rely on this assumption, which may ultimately lead to harming the usefulness of the framework. In this case, it is a shortcoming of the algorithms of the implementation, not of the conceptual framework, but still, it should be commented on in the guidelines. There are manners to address this, such as using non-convex clustering methods, which could be looked into in the future [6].

Another issue inherent to clustering is the K-problem, which talks about the fact that some clustering algorithms, such as the one used in this project (k-means), need to assume a number of clusters, k, beforehand. This can result in obfuscating the true underlying heterogeneous structure of the data. Solutions to this are multiple. One approach could be to allow the user to specify the k, and/or to perform creating, updating and deleting operations on the clustering results. Another approach would be to use a clustering algorithm that doesn't have this problem, such as Density-Based Clustering Algorithm (DBSCAN) [19], though it comes with other data-dependent parameters that may generate other limitations.

There were quite a few limitations that were gathered from the thematic analysis too, discussed in chapter 4. These suggest including guidelines to make the resulting framework inclusive, i.e. using a colour palette that is appropriate for colour-blind people. Another point that was hinted at multiple times was the difficulty of understanding the dimensionality reduction's purpose. This could be a matter of a lack of information in the information sheet used in the study, but it could also be solved by making it clearer in the guidelines what the purpose of it is.

Another noteworthy limitation is the sample size for the study. The time limitations and scope of this study didn't allow for a bigger sample size, which would have certainly aided in solidifying the results.

Lastly, another limitation stemming from the time limitation is the data types addressed by the framework. Even though the guidelines are generic and indeed model- datatype- and task- agnostic, the implementation was only made using time series data and for the regression task. It would further reinforce the results if the implementation was compatible with more data types, such as images or tabular data and could evaluate the results of classification. This could be something to take into account for future studies.

6 Conclusion

In this study we set out to prove the usefulness of a model- data-type- and task- agnostic framework based on clustering to perform machine learning performance analysis. To this end, we laid out the guidelines that constitute this conceptual framework. Then, we implemented the framework for time series regression, and tested it through a TAP study, a survey and two case studies. The results of the latter suggested that it indeed is possible to use our data type-, task- and model-agnostic, clustering-based framework to compare, evaluate and understand model performance in a machine learning problem, while also being preferable to classical performance metrics. Still, there is much future work to do which can build on this work. The two main research trajectories we identify are: (1) use more interpretable clustering/dimensionality reduction techniques and (2) implement ClAIrify for other data-types and test these implementations on a bigger cohort.

Bibliography

[1] Khaled Abuhlfaia and Ed de Quincey. "Evaluating the usability of an e-learning platform within higher education from a student perspective". In: *Proceedings of the 2019 3rd International Conference on Education and E-Learning*. 2019, pp. 1–7.

[2] Bilal Alsallakh et al. "Visual Methods for Analyzing Probabilistic Classification Data". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1703–1712. DOI: 10.1109/TVCG.2014.2346660.

[3] Energy and. *Energy and Emissions per Value Added Database - Data product - IEA*. 2023. URL: https://www.iea.org/data-and-statistics/data-product/energy-and-emissions-per-value-added-database.

[4] Natalia Andrienko et al. *Visual Analytics for Data Scientists*. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-56146-8.

[5] Ahlem Assila, Houcine Ezzedine, et al. "Standardized usability questionnaires: Features and quality focus". In: *Electronic Journal of Computer Science and Information Technology* 6.1 (2016).

[6] Sushant Bhargav and Mahesh Pawar. "A review of clustering methods forming non-convex clusters with missing and noisy data". In: *IJCSE* 4 (2016), pp. 39–44.

[7] Alexei Botchkarev. "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006* (2018).

[8] Alexei Botchkarev. "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006* (2018).

[9] Leo Breiman. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3 (2001), pp. 199–231.

[10] Victoria Clarke, Virginia Braun, and Nikki Hayfield. "Thematic analysis". In: *Qualitative psychology: A practical guide to research methods* 3 (2015), pp. 222–248.

[11] Elias Dabbas. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required*. Packt Publishing Ltd, 2021.

[12] Hercules Dalianis. *Evaluation Metrics and Evaluation*. Springer International Publishing, 2018, pp. 45–53. DOI: 10.1007/978-3-319-78503-5_6.

[13] Hercules Dalianis and Hercules Dalianis. "Evaluation metrics and evaluation". In: *Clinical text mining: secondary use of electronic patient records* (2018), pp. 45–53.

[14] Sanjoy Dasgupta et al. "Explainable k-means and k-medians clustering". In: *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria*. 2020, pp. 12–18.

[15] Dennis Dingen et al. "RegressionExplorer: Interactive Exploration of Logistic Regression Models with Subgroup Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 25 (1 Jan. 2019), pp. 246–255. ISSN: 19410506. DOI: 10.1109/TVCG.2018.2865043.

[16] A. Endert et al. "The State of the Art in Integrating Machine Learning into Visual Analytics". In: *Computer Graphics Forum* 36 (8 Dec. 2017), pp. 458–486. ISSN: 14678659. DOI: 10.1111/cgf.13092.

[17] A. Endert et al. "The State of the Art in Integrating Machine Learning into Visual Analytics". In: *Computer Graphics Forum* 36 (8 Dec. 2017), pp. 458–486. ISSN: 14678659. DOI: 10.1111/cgf.13092.

[18] Alex Endert et al. "The state of the art in integrating machine learning into visual analytics". In: *Computer Graphics Forum*. Vol. 36. 8. Wiley Online Library. 2017, pp. 458–486.

[19] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[20] Jean-Daniel Fekete. "Visual Analytics Infrastructures: From Data Management to Exploration". In: *Computer* 46.7 (2013), pp. 22–29. DOI: 10.1109/MC.2013.120.

[21] Kraig Finstad. "The usability metric for user experience". In: *Interacting with computers* 22.5 (2010), pp. 323–327.

[22] Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. "ExKMC: Expanding Explainable *k*-Means Clustering". In: *arXiv preprint arXiv:2006.02399* (2020).

[23] Daniel Gilmore et al. ""Giving the patients less work": A thematic analysis of telehealth use and recommendations to improve usability for autistic adults". In: *Autism* 27.4 (2023), pp. 1132–1141.

[24] Margherita Grandini, Enrico Bagli, and Giorgio Visani. "Metrics for Multi-Class Classification: an Overview". In: (Aug. 2020). URL: http://arxiv.org/abs/2008.05756.

[25] Margherita Grandini, Enrico Bagli, and Giorgio Visani. "Metrics for multi-class classification: an overview". In: *arXiv preprint arXiv:2008.05756* (2020).

[26] Guy S Handelman et al. "Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods". In: *American Journal of Roentgenology* 212.1 (2019), pp. 38–43.

[27] Fred Hohman et al. "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers". In: *IEEE Transactions on Visualization and Computer Graphics* 25 (8 Aug. 2019), pp. 2674–2693. ISSN: 19410506. DOI: 10.1109/TVCG.2018.2843369.

[28] Shammamah Hossain. "Visualization of Bioinformatics Data with Dash Bio". In: *Proceedings of the 18th Python in Science Conference*. Ed. by Chris Calloway et al. 2019, pp. 126–133. DOI: 10.25080/Majora-7ddc1dd1-012.

[29] Mohammad Hossin and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations". In: *International journal of data mining & knowledge management process* 5.2 (2015), p. 1.

[30] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[31] Minsuk Kahng et al. "A cti v is: Visual exploration of industry-scale deep neural network models". In: *IEEE transactions on visualization and computer graphics* 24.1 (2017), pp. 88–97.

[32] Trupti M Kodinariya, Prashant R Makwana, et al. "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6 (2013), pp. 90–95.

[33] Josua Krause, Adam Perer, and Enrico Bertini. "Using visual analytics to interpret predictive machine learning models". In: *arXiv preprint arXiv:1606.05685* (2016).

[34] Josua Krause, Adam Perer, and Kenney Ng. "Interacting with predictions: Visual inspection of black-box machine learning models". In: Association for Computing Machinery, May 2016, pp. 5686–5697. ISBN: 9781450333627. DOI: 10.1145/2858036.2858529.

[35] Todd Kulesza et al. "Principles of Explanatory Debugging to personalize interactive machine learning". In: vol. 2015-January. Association for Computing Machinery, Mar. 2015, pp. 126–137. ISBN: 9781450333061. DOI: 10.1145/2678025.2701399.

[36] Shixia Liu et al. "Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective". In: (Feb. 2017). URL: http://arxiv.org/abs/1702.01226.

[37] Shixia Liu et al. "Towards better analysis of machine learning models: A visual analytics perspective". In: *Visual Informatics* 1.1 (2017), pp. 48–56.

[38] Hossin M and Sulaiman M.N. "A Review on Evaluation Metrics for Data Classification Evaluations". In: *International Journal of Data Mining & Knowledge Management Process* 5 (2 Mar. 2015), pp. 01–11. ISSN: 2231007X. DOI: 10.5121/ijdkp.2015.5201.

[39] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.

[40] James Manyika et al. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011.

[41] Dastan Maulud and Adnan M Abdulazeez. "A review on linear regression comprehensive in machine learning". In: *Journal of Applied Science and Technology Trends* 1.4 (2020), pp. 140–147.

[42] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[43] Yao Ming et al. "Understanding hidden memories of recurrent neural networks". In: *2017 IEEE conference on visual analytics science and technology (VAST)*. IEEE. 2017, pp. 13–24.

[44] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. "A multidisciplinary survey and framework for design and evaluation of explainable AI systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11.3-4 (2021), pp. 1–45.

[45] Thomas Mühlbacher et al. "Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1643–1652. DOI: 10.1109/TVCG.2014.2346578.

[46] Meinard Müller. "Dynamic time warping". In: *Information retrieval for music and motion* (2007), pp. 69–84.

[47] openai. *GitHub - openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision*. June 2023. URL: https://github.com/openai/whisper.

[48] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[49] Alec Radford et al. "Robust speech recognition via large-scale weak supervision". In: *arXiv preprint arXiv:2212.04356* (2022).

[50] Sebastian Raschka. "Model evaluation, model selection, and algorithm selection in machine learning". In: *arXiv preprint arXiv:1811.12808* (2018).

[51] Paulo E Rauber et al. "Visualizing the hidden activity of artificial neural networks". In: *IEEE transactions on visualization and computer graphics* 23.1 (2016), pp. 101–110.

[52] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[53] Bahador Saket, Alex Endert, and Çağatay Demiralp. "Task-Based Effectiveness of Basic Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 25.7 (2019), pp. 2505–2512. DOI: 10.1109/TVCG.2018.2829750.

[54] David Schultz and Brijnesh Jain. "Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces". In: *Pattern Recognition* 74 (2018), pp. 340–358.

[55] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. "A review of supervised machine learning algorithms". In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Ieee. 2016, pp. 1310–1315.

[56] Thilo Spinner et al. "ExplAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning". In: *IEEE Transactions on Visualization and Computer Graphics* 26 (1 Jan. 2020), pp. 1064–1074. ISSN: 19410506. DOI: 10.1109/TVCG.2019.2934629.

[57] Hamed Taherdoost. "What is the best response scale for survey and questionnaire design; review of different lengths of rating scale/attitude scale/Likert scale". In: *Hamed Taherdoost* (2019), pp. 1–10.

[58] Gary K. L. Tam, Vivek Kothari, and Min Chen. "An Analysis of Machine- and Human-Analytics in Classification". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 71–80. DOI: 10.1109/TVCG.2016.2598829.

[59] Romain Tavenard et al. "Tslearn, A Machine Learning Toolkit for Time Series Data". In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-091.html.

[60] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[61] Ohio University of Dayton. *Average Daily Tempreature Archive*. 2023. URL: https://academic.udayton.edu/kissock/http/Weather/citylistWorld.htm.

[62] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[63] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[64] Kanit Wongsuphasawat et al. "Visualizing dataflow graphs of deep learning models in tensorflow". In: *IEEE transactions on visualization and computer graphics* 24.1 (2017), pp. 1–12.

[65] Junjie Wu and Junjie Wu. "Cluster analysis and K-means clustering: an introduction". In: *Advances in K-Means clustering: A data mining thinking* (2012), pp. 1–16.

[66] Jason Yosinski et al. "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579* (2015).

[67] Jun Yuan et al. *A survey of visual analytics techniques for machine learning*. Mar. 2021. DOI: 10.1007/s41095-020-0191-7.

[68] Jiawei Zhang et al. "Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models". In: (Aug. 2018). DOI: 10.1109/TVCG.2018.2864499. URL: http://arxiv.org/abs/1808.00196%20http://dx.doi.org/10.1109/TVCG.2018.2864499.