# Clustering small and medium sized Dutch enterprises using hybrid intelligence

Shipra Sharma

TU Delft

# Clustering small and medium sized Dutch enterprises using hybrid intelligence

by

Shipra Sharma

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 23, 2021 at 02:00 PM (CET).

| | | |
|---|---|---|
| Student number: | 5093406 | |
| Project duration: | November 19, 2020 – August 23, 2021 | |
| Thesis committee: | Prof. dr. ir. Geert-Jan Houben | TU Delft, chair |
| | Dr. ir. Jie Yang | TU Delft, supervisor |
| | Dr. ir. Cynthia Liem | TU Delft, committee member |
| | Ir. Estelle Rambier | Exact, external expert |

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Picture on cover taken from
`https://bdtechtalks.com/2020/03/04/gary-marcus-hybrid-ai/`.

TU Delft = exact

# Disclaimer

*The information and data made available by Exact is under strict confidentiality and is to be used for the research specific to this thesis work only.*

# Preface

Here I present to you my graduation thesis titled "Clustering small and medium-sized Dutch enterprises using Hybrid Intelligence". This thesis is written to obtain the Master of Science degree from Delft University of Technology (TU Delft). The project was conducted in collaboration with Exact, Delft. I completed my research and application work under the supervision of Dr. Ir. Jie Yang (TU Delft), Ir. Sihang Qiu (TU Delft), and Ir. Estelle Rambier (Exact), from November 19, 2020, until August 23, 2021.

It seems like yesterday when I first visited the campus during TU Delft's IP, 2019, and here I am writing this thesis to conclude the journey. There we were, all excited about life's new phase, adapting to the rains and winds, attending lectures in halls Ampere and Boole in the famous EWI building, struggling through group projects and earning credits, and enjoying the CH lunch lectures. Life seemed all good but then Covid-19 hit us and put a stop to the normal life, but with TU Delft's online support everything became accessible quickly, and "being virtual" became the new normal. Now, this phase of my life is about to end and I am looking forward to all my future endeavors. But before we conclude this chapter of my life, I would like to acknowledge the people who helped me reach the point where I am today.

Firstly, Dr. Ir. Jie Yang, Ir. Estelle Rambier and Ir. Sihang Qiu, I sincerely thank you for all the guidance. Thank you for taking out all the time for the numerous meetings and brainstorming sessions we had to help me achieve this. I wish you all good luck with all your future goals and life milestones. I thank you, Prof. Dr. Ir. Geert-Jan Houben and Dr. Ir. Cynthia Liem for being a part of my thesis committee. I am also thankful to the entire Data Science team at Exact, who guided me throughout to understand the company's tasks better.

Last but not the least, I would also like to use this platform to thank my parents, my brother, and my friends, for all those times when they believed in me and supported me in all the ways they could. I am grateful to you all for motivating me and lifting me whenever it all seemed impossible, especially during the uncertain Covid times.

Now is the time to end this chapter of my life and look forward to the opportunities and experiences the future holds for me!!

*Shipra Sharma*
*Delft, August 2021*

# Abstract

We are surrounded by data. Data capture the characteristics of any entity around us, like living species, properties of scientific experimentation, etc. Moreover, data helps one to make intuitive decisions about advance analysis, or decision-making. One of the most common applications of data analysis is to create groups of similar structure to understand the underlying structures of a given data set. The classification system is either supervised or unsupervised, depending on whether the data objects are assigned to well defined labels or undefined categories. In unsupervised classification or clustering there are no labels target labels available.

State-of-the-art clustering algorithms are developed in a broader sense without targeting any specific applications. However, they are used in a multiple application domains. Since these algorithms lack domain-specific information and user-specific input interface, they might not always produce user specific results. Also, categorical features in the dataset make clustering harder because they lack semantics. Moreover, unlike classification tasks which are validated using specific class labels, clustering is subjective because it depends on the interpretation, and user requirements. The little notion of ground truth makes cluster validation harder in an unsupervised setting. Also, there is no universally adopted approach to choose features or clustering schemes.

To tackle such challenges there is an increased demands of devising new clustering approaches where the humans are used in the clustering analysis workflows to make clustering results domain specific and qualitative. Such an approach where we try to achieve better computation results using human knowledge lies under the hybrid intelligence domain.

This thesis explores the possibility of designing an end-to-end clustering analysis workflow using hybrid intelligence. The thesis aims to answer the following research question - *How can we use hybrid intelligence in cluster analysis workflow to generate user-specific clusters and evaluate them?* We try to answer the research question by introducing multiple novel methods that aim to solve the following challenges: evaluating clusters, creating semantics in categorical features, and performing user-specific cluster analysis. We apply the developed methodologies on real-time financial data to cluster small and medium (SMEs) sized Dutch enterprises. By our experimentation, we can observe that we manage to cluster Dutch SMEs as per the user-specific goals based on their financial behaviors. We believe that we achieve such results by creating semantics in the categorical features. The clustering results are further analyzed from a user requirement perspective. Our proposed cluster validation game enables us to validate cluster objects using human intelligence. The associated experimentation results give us confidence in our hypothesis that hybrid intelligence is one of the solutions to solve the clustering challenges.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

We are surrounded by data. Data capture the characteristics of any entity around us, like living species, properties of scientific experimentation, etc. Moreover, data helps one to make intuitive decisions about advance analysis, or decision-making. One of the most common applications of data analysis is to create groups of similar structure to understand the underlying structures of a given data set. Data objects in a particular cluster should have similar properties based on some criteria. For example, all biological beings are classified into two main categories - animals and plants, and animals are then further classified into different groups like human beings, tigers, lions, etc. depending on their characteristics[68]. The classification system is either supervised or unsupervised, depending on whether the data objects are assigned to well defined labels or undefined categories. In unsupervised classification, also called, clustering no labeled data is available, i.e there is no notion of the classification classes[26]. The goal of clustering is to explore the unknown structures of data from very less or no prior information.

Generally, state-of-the-art clustering algorithms are developed in a broader sense without targeting any specific applications. However, they are used in a multiple application domains. Since these algorithms lack domain-specific information and user-specific input interface, they might not always produce user specific results[3]. Human interaction in cluster analysis can be used for various purposes like

- Improving clustering quality: The most popular goal to use humans in the loop is to enhance the quality of clustering results[3]. Chuang et al.,[20] believe that this is the sole purpose of involving humans in the clustering process. Most of the researchers believe that the merged capabilities of humans and machines would produce better results than a complete manual or automated process [5, 8, 12, 19].

- Understanding the clustering process: Another reason to add humans in the loop is to properly understand the approach by which the machines computer clusters [11, 22, 33].

- Transforming and extracting interesting features from the data: There are various types of data out of which categorical data, as per our hypothesis, can be the one that might leverage the benefits of human involvement. Categorical data clustering refers to the case where the data objects are to be grouped over categorical attributes. "A categorical attribute is an attribute whose values come from a set of discrete values that are not inherently comparable. That is, there is no objective ordering or inherent

distance function for the categorical values, and there is no mapping from categorical to numerical values that is semantically sensible[2]". As explained earlier, data clustering targets to group objects based on some similarity measure. This tells us that there should be a way for the machines to judge this notion of similarity. Most of the clustering algorithms in the literature pay attention to clustering data objects that have numerical attributes. Because in case of the numerical data, the similarity can be easily derived using numerical comparisons. For example, it is easier to conclude that the values 10000 and 11349 are more similar than 10000 and 333 based on mathematical understanding. But not all data sets have numerical attributes. E.g., the data in table 1.1 represents the outstanding amount that an organization needs to pay to its employees:

- 'Employee ID' which represents the unique ID representing an employee.
- 'Project ID' represents the project for which an employee works.
- 'Department' represents the organization's department to which an employee belongs.
- 'Outstanding Amount' which represents how much amount is to be paid to an employee.

| Employee ID | Project ID | Department | Outstanding Amount (Euros) |
|---|---|---|---|
| 01 | 33 | Accounts | 279 |
| 02 | 21 | IT | -780 |
| 03 | 15 | Support | 1236 |

Table 1.1: An example of a categorical dataset

From the above example, we can observe that a few of the data points are categorical attributes like 'Project ID' and 'Department' and 'Outstanding Amount (Euros)' is numerical. Such categorical features can not be compared as they represent values of certain categories. It is not directly obvious that how does one defines the similarity between the values such as 'Accounts' and 'IT' or 'Project ID' 33 or 21. Also, we need insight into what exactly does '-' sign represent?
Similarly, there are ample examples of categorical data like pharmaceutical data, where medicines are defined over attributes such as chemicals, quantity, or side-effects; employees data, where individual's information includes attributes such as age, gender, discipline, address, etc. Thus, there is a wide range of applications where one might need to cluster and analyze clusters using categorical values as well.

- Defining the subjectivity of the target clustering goal: Many clustering algorithms can be used for the same data to generate clusters, and there is no definitive approach to choose one. But, the end-users have certain expectations and the goal of involving humans in the clustering process then is to find the most suitable clustering model/approach that tries to fulfill defined goals[17, 24, 67].

To experiment with the feasibility of involving humans in the clustering process to achieve any of the above-mentioned benefits, several related work has been performed where humans and machines work together to produce relevant clustering results. Researchers try

to engage humans in the clustering process to make it more domain specific and meet the user requirements. Related work for designing such human-in-the-loop clustering techniques has been explored by several research communities including data mining, visual analytic, machine learning, and human computation[3].

## 1.1. Thesis Goal

As we discussed above, researchers in various communities including data mining, visual analytic, machine learning, and human computation are working to build systems that interact with humans to carry out the clustering task more efficiently. Motivated by the same, this thesis explores the possibility of designing an end-to-end clustering workflow using humans in the loop. This workflow aims to generate relevant and user-specific clusters within an application domain. However, there are multiple scientific challenges that a user can face while clustering data objects which are discussed elaborately in the next section. We hypothesize that the relevant clusters can be obtained by feeding good feature-engineered data to the machine learning algorithms and by tuning the algorithm computations with approaches like adjusting a similarity measure[3]. We aim to use human knowledge during the entire cluster analysis workflow, for example, the initial steps like feature engineering, data transformation, etc. for converting a categorical data set to an inherent numerical data set, so that we entirely skip the challenges of categorical data clustering. We also intend to use human understanding of a domain to validate the relevancy of clusters and to validate our feature engineering choices in terms of cluster requirements. The validation method aims to provide some confidence over the clustering results. This human-based validation should be able to assess the relevancy of cluster objects irrespective of which clustering algorithm we use. The proposed workflow aims to provide insightful answers to questions like 'how many clusters would be enough' or 'is clustering algorithm generating required clusters?'.

## 1.2. Scientific Challenges

From the above discussion, one can encounter the following scientific challenges while trying to cluster data objects:

- State-of-the-art clustering algorithms are developed in a broader sense without targeting any specific applications. However, they are used in a multiple application domains. Since these algorithms lack domain-specific information and user-specific input interface, they might not always produce user specific results. Also, categorical features in the dataset make clustering harder because they lack semantics. Thus, one needs to derive semantics and feature engineer the data points accordingly, via a deeper understanding of the categories and their respective values. In addition, several other inputs are required for clustering, for example, the number of clusters, etc. Thus, it is not very feasible to conduct the entire cluster analysis automatically. Also, the larger amount of data doesn't allow a complete manual process as well.

- unlike classification tasks which are validated using specific class labels, clustering is subjective because its quality or relevancy depends on the user requirements. The little notion of ground truth makes cluster validation harder in an unsupervised setting. Also, there is no universally adopted approach to choose features or clustering

schemes[68]. Also, a large part of "understanding" is to understand the approach how machines or algorithms compute clusters. Thus, we need effective validation standards that authenticate our feature selection and engineering choices, or clustering algorithm and their hyperparameters choices to observe whether the generated clusters meet the clustering demands.

## 1.3. Research Questions

To deal with the above-mentioned challenges we need to come up with methodologies that generate user-specific clusters using human intelligence. Also, we need to design an effective cluster validation scheme that validates the choices and steps that we take for cluster generation. Therefore the main research goal of this study is to implement an end-to-end cluster analysis workflow to leverage the benefits of involving humans in a loop. To achieve this goal, the following research question needs to be answered:

- *RQ: How can we use hybrid intelligence in cluster analysis workflow to generate user-specific clusters and evaluate them?*

We try to find the answer for the above-mentioned research question by finding solutions for the following sub-questions:

- *SQ1: What is the state of the art in data clustering and hybrid intelligence approaches?*
  To answer this question, a literature study is conducted in which the related work in hybrid intelligence and cluster analysis is discussed. This study also aims to find an intersection of the two domains which further motivates our work of using humans in the loop to implement clustering more effectively.

- *SQ2: How and where to use hybrid intelligence in cluster analysis workflow?*
  This question involves the methodology and system design to include humans in the loops to generate relevant clusters by using human knowledge in the various steps of cluster analysis i.e. data transformation, feature engineering and feature extraction, clustering model selection, and validating the generated clusters using human inputs.

- *SQ3: How to use the proposed hybrid intelligent clustering workflow to generate customer clusters from financial data?*
  To answer this question, we apply our workflow to a real-time use case where we try to cluster a company's customers based on their financial transactions.

Chapter 2, chapter 3 and chapter 4 provide the detailed approach and answers to the sub-questions.

## 1.4. Contributions

Through the answers provided for the above-mentioned sub-questions, the contributions of this thesis can be summarized as follows:

- C1: We contribute a literature survey on clustering, hybrid intelligence, and their intersection. We first look into the state-of-the-art methodologies to define clustering,

its types, applications, and challenges. We then investigate the applications of hybrid intelligence to solve the clustering challenges. These findings serve as our motivation to further design our methodologies.

- C2: We design a human computational workflow that tackles the challenges of data clustering. This workflow includes four novel methods which aim to standardize the requirement and knowledge collection steps, create semantics for categorical features, perform domain-specific cluster analysis and evaluate cluster objects using humans.

- C3: Finally we apply our design to a real-time use case where we try to cluster a company's customers based on their financial data. The experimentation steps under this contribution give us a chance to assess the feasibility and efficiency of our proposed methods. It also helps us investigate the fact that to which extent were able to solve the above-mentioned clustering challenges.

## 1.5. Thesis Outline

The complete structure of this report is as follows: In Chapter 2 we make an extensive review of the literature covering topics on clustering, hybrid intelligence, and its applications and sub-applications. This chapter provides the answer to SQ1. Chapter 3 introduces the system design of our proposed end-to-end human computational cluster analysis workflow. In Chapter 4, we apply our workflow to financial data to validate the effectiveness of our end-to-end human computational cluster analysis workflow in a real-time use case. The chapter is concluded by discussing the workflow's limitations. In Chapter 5 we evaluate the different components of our workflow via multiple evaluation metrics like accuracy or cognitive load. Chapters 3, 4, and 5 collectively provide answers to SQ2 and SQ3. Chapter 6 concludes the report by answering the research questions. It also includes the future work that could be built on using the results of this thesis work.

# Chapter 2

# Related Work

This chapter aims to provide the answer to *SQ1: What is the state of the art in data clustering and humans in the loop approaches?* It presents an exhaustive literature survey around clustering and hybrid intelligence. This survey helps in understanding clustering and its applications. The chapter also covers the existing applications and sub-applications of hybrid intelligence, which serve as a motivation to further design our methodologies.

## 2.1. Clustering

Clustering has been an component of data analysis for a very long time now. We live in a world where the volume of data increases exponentially on a daily basis. Clustering plays a crucial part in understanding the underlying behaviours of the collected data. Hence in the recent years, a lot of researches have taken place to make clustering better to handle larger amount of data and make it more efficient. Xu et al.,[68] defines a cluster as "continuous regions of a d-dimensional feature space containing a relatively high density of points, separated from other such regions containing a relatively lower density of points". Literature suggests that the data objects of same clusters should be similar and those belonging to different clusters should be dissimilar. They also lay emphasize that how well one should objectively define the related (dis)similarity [31, 34, 39]. Over the past decades, Clustering has had multiple applications like machine learning, pattern recognition, web mining, database analysis, information retrieval, life, and medical sciences, astronomy and earth sciences, marketing, financial businesses, and so on. Thus it can be easily deduced that clustering is omnipresent.

Cluster analysis can be achieved through four basic steps[68]:

1. Feature Selection or extraction: Jain et al.,[47] and Bishop et al.,[7] describe that feature selection incorporates choosing distinct features from a data set whereas, feature extraction aims to perform transformations on the available features to deduce useful and novel features from the original ones present in a data set. In our case, where we aim to work with categorical data, feature extraction could be used to generate semantic features to better understand the underlying structure of the data.

2. Clustering algorithm selection: A diverse range of clustering algorithms have been developed to tackle the clustering problems depending on the goals. Therefore, it is important to carefully investigate the requirements of a problem to choose a fitting

algorithm. For example, *K-means* algorithm is based on similarity distances like Euclidean distance and hence results in hyper-spherical clusters. However, in a given dataset if the underlying clusters are in some other forms *K-means* might not work. Thus, choosing a relevant clustering algorithm is highly important concerning the data set features and clustering goal [68].

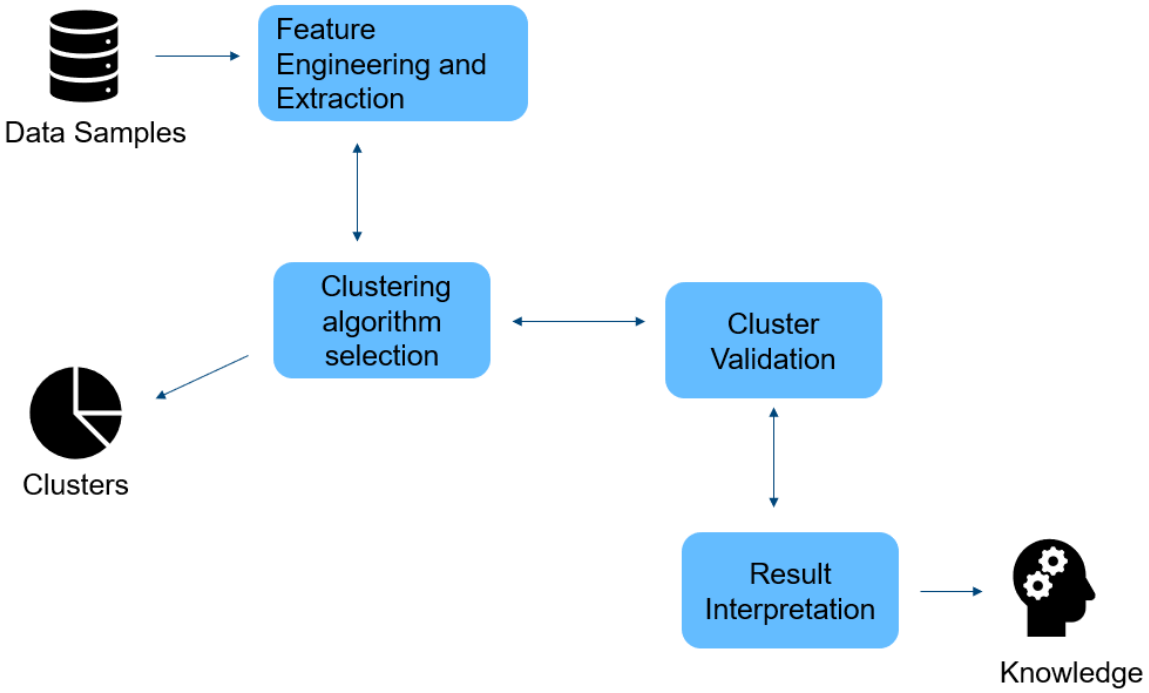



Figure 2.1: 4 steps of cluster analysis

3. Cluster Validation: Given an input data set, a clustering algorithm will produce clusters whether or not they exist in the data. Thus, it is highly important to have an objective assessment of the obtained clusters to verify the correctness of the clusters obtained.

4. Result Interpretation: The end goal of clustering is to produce meaningful insights for the end-users from a given data set. This is so because users want to develop a clear understanding of the data to solve further problems like business predictions, risk analysis, etc. Thus it is highly important to interpret the results correctly in the forms of visuals or statistics which could be used as an input for suggesting hypotheses of further machine learning problems.

It is important to know that cluster analysis is not a one-shot solution. It is iterative (as shown in figure 2.1, we can move in both the directions from any given step), and depending on the output of one of the steps, we can modify the other steps to obtain as relevant results as possible. This iterative requirement serves as our motivation to use human intuitions and knowledge to analyze and modify the results of different steps, to achieve a good result at the end.

### 2.1.1. Data Types

Almost any information can be represented as data. Understanding the type of data is a crucial step for exploratory data analysis as different data types require different processing[1, 39]. Most of the data points can be categorized into 4 basic types from a Machine Learning perspective: numerical data, categorical data, time-series data, and text.

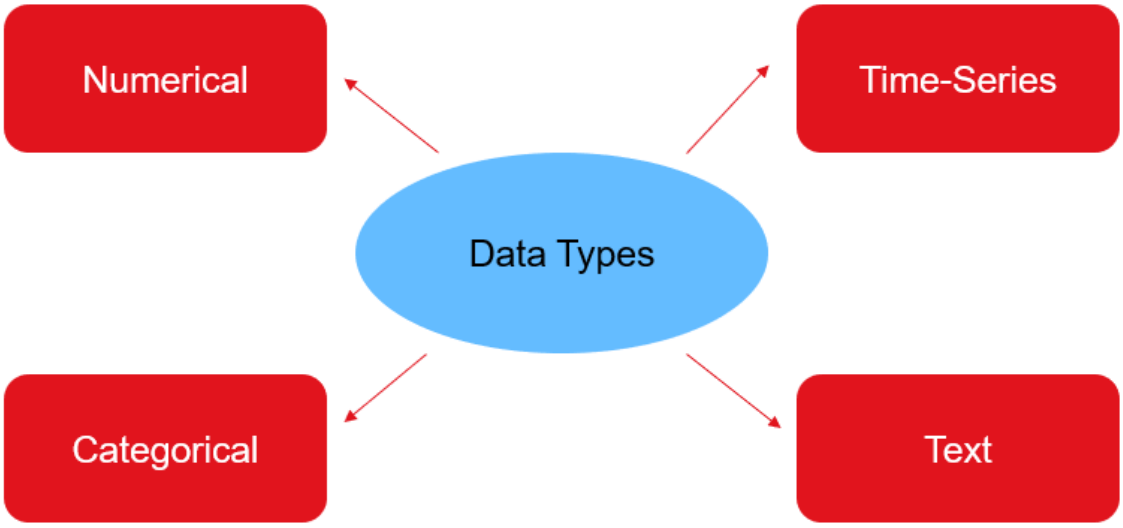



Figure 2.2: Data types in machine learning

- Numerical Data: any data points which are exact numbers fall into the category of numerical data. This data has meaning as a measurement such as weight and age of an individual or revenue generated by a company in a financial year. Numerical data can be further characterized as continuous data or discrete data. Continuous data can have any value within a range whereas discrete data has distinct values. For example, the number of houses sold by a real estate company are discrete numbers, such as 20, 99, 100, etc, and can never be 2.5, 7.89, etc. Moreover, values like age and weight of an employee can lie in a range for example 81.5 kilograms and 23.7 years[27].

- Categorical Data: Categorical data represents characteristics, such as a student's name, gender, or hometown. Categorical data can take numerical values as well. For example, in table 1.1 we use 33 or 21 as 'Project ID'. But these numbers don't have a mathematical meaning. Thus, adding them or taking an average won't help us in finding any meaning. Numerical data and categorical data can be combined in some instances and form ordinal data. In ordinal data, the categories are represented as ranked numbers. For example, in a survey form, the user satisfaction for a service can be measured on a scale from 1 to 10, where 1 represents the least satisfaction and 10 represents the most satisfaction[27].

- Time-Series Data: Time series data contains different values of for a variable, taken at a different time interval. For example, the room temperature may have different

values at different times of the day. Such data is used to make business decisions like sales predictions, market growth or downfall, etc.

- Text Data: Data represented as statements, paragraphs or documents is what text data is. Examples of a text data set are students' essays, tweets, articles, etc. Natural Language Processing and Information Retrieval domains generally target text data problems.

### 2.1.2. Types Of Clustering

There are multiple approaches to perform cluster analysis depending on the type of data and the problem that we aim to solve.

**Partitional Clustering**

"Partitional clustering is a clustering method used to classify observations, within a data set, into multiple groups based on their similarity. The algorithms require the analyst to specify the number of clusters to be generated. Partitional algorithms determine all clusters at a single time"[51]. The various algorithms of partitional clustering are as follows:

- "K-means clustering, in which, each cluster is represented by the center or means of the data points belonging to the cluster. The K-means method is sensitive to anomalous data points and outliers"[50].
- "K-medoids clustering or PAM, in which, each cluster is represented by one of the objects in the cluster. PAM is less sensitive to outliers compared to k-means"[41].
- "CLARA algorithm (Clustering Large Applications), which is an extension to PAM adapted for large data sets"[71].

Among various types of partitional clustering algorithms, K-Means is one of the most popular algorithms[69].

**Hierarchical Clustering**

"Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This iterative process continues until all the clusters are merged. Hierarchical algorithms find successive clusters using previously established clusters. Hierarchical algorithms can be further agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them in successively larger clusters"[51].

**Clustering based on the data type**

1. Numerical Data clustering: Clustering algorithms for numerical data points can be based on multiple techniques like hierarchical, partitional, or density-based techniques[27, 51, 68]. Hierarchical clustering techniques treats every data point as a single cluster and iteratively merge similar data points into one cluster based on the measures like cosine distance or euclidean distance. They repeat the merging steps until the clustering requirement is met [40]. "Partitional clustering techniques often start with an

initial, usually random, partition and proceed with its refinement by locally improving the optimization criterion. The majority of them could be considered greedy-like algorithms. They suffer from the fact that they can easily get stuck to local optima"[13]. "Density-based clustering algorithms work by detecting concentrated areas, empty areas, or sparse areas concerning data points. The data points that are not part of a cluster are labeled as noise"[18].

2. Categorical Data Clustering: As discussed earlier, for categorical data it is hard to measure data points' similarity. "The well-known measures such as Hamming distance [39] which measures the number of common values between two tuples, or the Jaccard similarity measure, which is defined as the intersection over the union of the values in the two tuples are based on this approach". Some algorithms which are based on this approach are *K-modes*[38] and *ROCK*[32]. Several algorithms like *CACTUS*[28] and *STIRR*[53] uses context-based similarity measures where relationships between categorical values are defined by comparing the context in which they appear. The more similar these two contexts are, the more similar the attribute values are. For example in table 1.1 the *Department* values *Accounts* and *Support* can be used in same context as they both represent positive values for *Outstanding Amount*, however *IT* might have different context as it represents negative *Outstanding Amount* value.

## 2.2. Hybrid Intelligence

Hybrid Intelligence aims to solve tasks that are difficult for machines alone like rating the quality of a website[63]. The use of human intelligence was early embraced in machine learning for purposes like image annotations for better computer vision algorithms or for creating labels for supervised learning. Such an approach where we try to achieve better computation results using human knowledge lies under the hybrid intelligence domain. Hybrid intelligence takes human expertise and intentionality into account while the computations and actions are performed by the machines. The research advanced in the field with multiple applications as stated below:

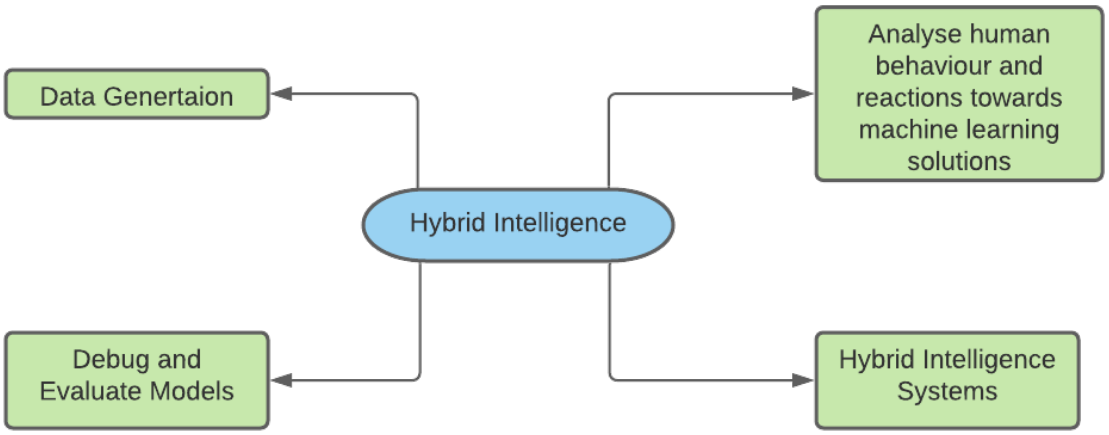



Figure 2.3: A few applications of Hybrid Intelligence

- As per our research, the most common way of using human intuitions is generating labels[63]. These labels can either be binary or categorical. Binary labels can have only two possible values, for example, true-or-false or yes-or-no. The categorical labels can have multiple values like weather labels - summer, winter, spring, or fall[9, 16]. Von Ahn et al.,[66] mention how they use human intelligence to generate a transcription of printed text. Humans have been used in other tasks like the translation of a sentence from one language to another[10] or generating image annotations[44].

- Another application of using human intelligence is to debug and evaluate the coherence of a topic model or exploring thematic topics discussed in a set of documents. Chang et al.,[14] propose a way to use human intelligence to measure the quality of a set of topics in an unsupervised learning setting. They do so by proposing a *human intrusion task* where a crowd-worker is presented with a list of common words from a topic and one *intruder* (i.e. a word that seems less common to a topic). The crowd-worker then assigns higher weights to the words that seem related and lower weight to the intruder. They observe that this is easier if the topic is coherent and harder otherwise. Some work has also been done in developing models that are human-interpretable, especially in sensitive domains like health and criminal justice where a user needs to understand the model's prediction[23, 43, 61].

- Human intelligence is also used to analyze humans' reactions towards machine learning systems. These reactions are subjective in nature and often require an understanding of why humans react to a certain situation in certain manner. Literature reveals multiple examples of behavioral studies aimed at understanding user trust in algorithmic predictions[21, 25, 56] and how users react to applications like online advertising[29].

- Hybrid Intelligence Systems is another example where one includes humans in loops to perform tasks that rely on human intelligence or knowledge. Such hybrid systems can perform better than humans or machines alone[63]. Various tasks that can be achieved by hybrid intelligence systems are grading students' work[46], writing essays or novels[42], building better topics models[14], transcribing speech in real time[54], scheduling conference sessions[6],forecasting geopolitical or economic events[4], and hybrid clustering.

### 2.2.1. Hybrid Clustering

Since our work focuses on clustering thus, now we explore hybrid clustering in detail. Hybrid intelligence systems can be used in scenarios where understanding data points is easier for humans, like in our case understanding the categorical data points. For example, given a data set of companies' logo images, a human could use their life experience and knowledge to categorize them into sets like "IT, “Entertainment”, etc. while a machine without access to a database can not[63]. Following approaches are encountered in the literature around hybrid clustering:

- Hybrid clustering techniques have been proposed which include human comparisons to generate/modify a similarity matrix or other similarity function. For example, Tamuz et al.,[60] designed an adaptive algorithm that estimates a similarity ma-

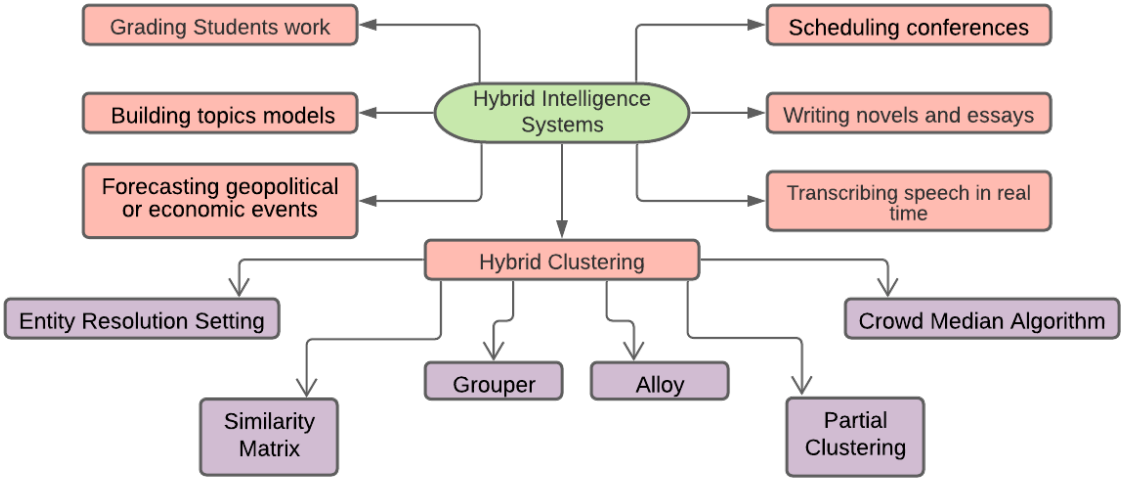



Figure 2.4: A few applications of Hybrid Intelligence Systems

trix from human judgments based on comparisons of triples ("Is object X more similar to object Y or object Z?"). This approach requires only a relatively small number of human judgments to obtain a good approximation. While studying the challenges of data clustering, this was one of the major challenges encountered in that there is no general notion of similarity. Similarity depends on the context of the problem that one is dealing with. Thus, this work tries to solve the same challenge by defining similarity in the domain context.

- Some approaches pay attention to the entity resolution setting i.e. an object belonging to a similar entity should be clustered together [52, 64]. For example, a tiger, a lion, a horse can be clustered together as they belong to the entity *animals*, whereas a peacock, a nightingale, a hummingbird can be clustered together as they belong to the entity *birds*.

- Gomes et al.,[30] hypothesis that the absence of the entire context might result in unstandardized results from humans. They propose a Bayesian model to accumulate partial clusters obtained by chunking a larger dataset into smaller data object sets. Such partial clusters are formed when smaller sets of data objects are presented to crowd-workers and they are asked to cluster only the presented smaller set. The authors try to solve the challenges of the accumulation step and conclude that Bayesian crowd clustering works well. This literature proves that overall contextual understanding of the target goal plays an important role, especially when one needs standardization of the tasks.

- Antti et al.,[36] identify that the efficiency of human computation is based on the capabilities of humans to process information better than machines. However, they also agree that all machine learning algorithms rely on mathematical operations, such as sums, averages, least-squares, etc. that are less suitable for human computation. Thus, through their work, they put the effort into combining these two aspects of data processing. They do so by proposing a crowd median algorithm where the

crowd-worker is presented with *S* (number of clusters) and *x* (new object). They are then asked to identify which object in *S* is closest to *x*. They then apply the crowd median algorithm as follows: present three objects from S and ask which seems like an outlier, repeat these triples and the least chosen outlier becomes the center. This works like the centroids in the K- means clustering algorithm. Thus, the authors conduct experiments and validate their hypothesis that humans and machines can work simultaneously to create better results as compared to the results generated by humans or machines alone. This validation serves as a base for our hybrid model where we try to solve clustering challenges by using humans in the loop.

- Chang et al.,[15] state that Crowdsourced clustering approaches present an efficient way to process complex semantics from any information. However, they believe that the existing approaches have difficulties supporting the global context needed for workers to generate meaningful categories, and are costly because the majority of items require human judgments. To introduce the global context to crowd workers, they introduce *Alloy* which tries to the inherent advantage of humans over machines for the complex problem of understanding unstructured data beyond merely measuring surface similarity. It uses a hybrid of *Cast*(coworkers) and *Gather*(machine backbone). The *head cast* perform active sampling and searching to identify the global context of the entire dataset. They find see, highlight keywords and search and label the data set. They then use a hierarchical clustering algorithm as a machine backbone. The *merge cast* then identify duplicates, sub-categories, etc. in the labels selected by the *head cast*. The *tail cast* then go through the leftover keyboards and either add them to the existing clusters or make a new cluster. The model is evaluated using *Mutual Information Metric* to identify symmetric measurement between the number of clusters and precision of each cluster. Through their experimentation, the authors proved that *Alloy* manages to change the crowd's task from classifying a fixed subset of items to actively sampling and querying the entire dataset, which gives them a global context of the entire problem rather than just working on the subset. We believe that contextual knowledge is highly needed for any engineer or crowd worker to understand the task better and execute it more efficiently.

We can observe from all the literature mentioned in chapter 1 and 2 that a significant amount of work has been done in using humans in the clustering process loop for multiple objectives like understanding the data, improving cluster quality, creating a similarity notion, generating clusters using crowd-workers, etc. Motivated by this, we choose to design and experiment with an end to end human computational workflow which tries to use human intelligence at multiple places in the cluster analysis process pipeline like understanding and transforming data; clustering model selection, adjusting its hyperparameters for the most relevant cluster generation; and last but not the least validating the generated clusters.

## 2.3. Summary

Through this chapter, we got acquainted with clustering, its types, data types in machine learning, the approaches to perform data clustering. We familiarize ourselves with hybrid intelligence, its applications, and sub-applications. We witness that hybrid intelligence is

one of the possible domains that pays attention to overcoming the challenges of clustering. To yield the best possible clusters, one needs to tailor the clustering process as per the desired goals and expectations. Motivated by the need for domain-specific tailoring in clustering process and advancement and strengths of hybrid intelligence, we propose that one of the approaches of handling clustering challenges mentioned in section 1.2 can be served by using humans in cluster analysis workflow, to generate relevant clusters from a domain-specific dataset. We believe that this can be achieved by intuitively understanding the data and clustering goals, data transformation, feature engineering, clustering model selection, and their hyperparameters' tuning, and proper cluster interpretation.

# Chapter 3

# Methodologies

As explained in Section 1.1, this thesis explores the possibility of designing an end-to-end clustering workflow using humans in the loop. This chapter aims to find the answer to *SQ2: How and where to use hybrid intelligence in cluster analysis workflow?* In Chapter 2, we discussed the various steps of a state-of-the-art cluster analysis process. Based on that, we further break down SQ2 into sub-questions per the different steps of the cluster analysis process, to design our human computational cluster analysis workflow.

1. How can we design a standardized approach to gather user requirements for cluster analysis?
2. How do we add semantics to categorical features?
3. How do we use hybrid intelligence to analyze whether the generated clusters meet the user requirements?
4. How do we use hybrid intelligence to validate the clusters?

From the above questions, we can observe that we propose to use hybrid intelligence in multiple stages of clustering workflow. A workflow that uses humans, as well as machines to achieve a target goal, can be categorized as a hybrid intelligent system. "Such a system organizes humans and machines to carry out the process of computation — whether it be performing the basic operations, taking charge of the control process (e.g., decide what operations to execute next), or even synthesizing the program (e.g., by creating new operations and specifying how they are ordered)"[48]. Figure 3.1 shows the steps of our proposed workflow. The steps in color green show our proposed methodologies which are discussed elaborately in the later sections. The steps in color blue are the steps adapted from the state-of-the-art clustering analysis workflow which are explained in Chapter 2.

The above-mentioned 4 questions are answered one by one in the following sections.

## 3.1. Gathering User Requirements And Understanding Data

To start, we first need to address that why is understanding data crucial for any cluster analysis process. Understanding the given data that one intends to work on is a crucial step of any machine learning application. With a good understanding of data, one can make better decisions required for tasks like finding new customers, increasing customer retention, improving customer service, or predicting sales. Data helps one to find solutions

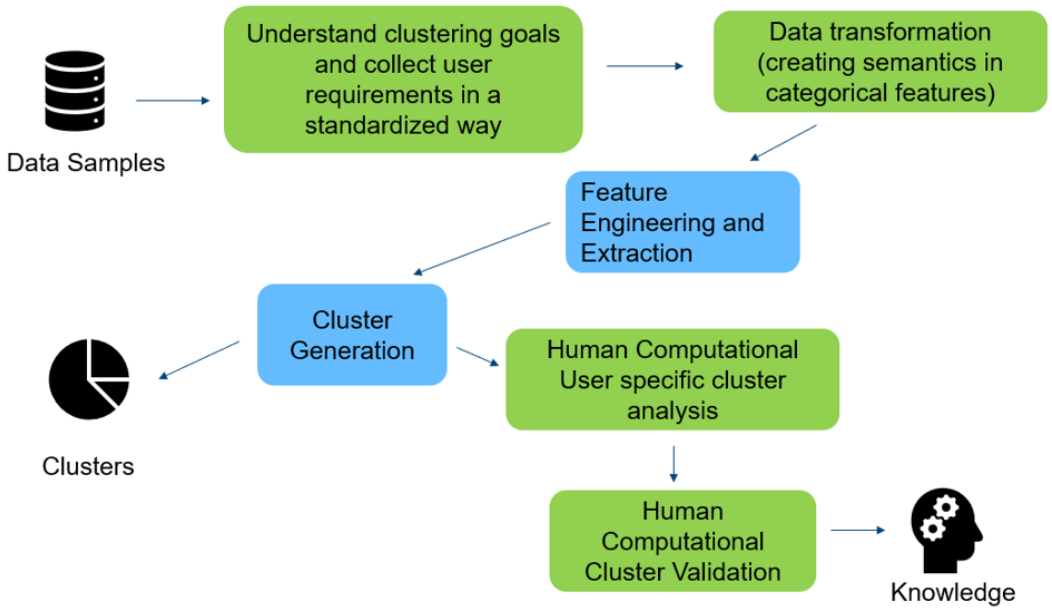



Figure 3.1: Cluster Analysis Steps in our proposed end-to-end Human Computational workflow

for business problems like low revenue generation. Data helps to improve the processes and last but not least data helps one to understand customers better[62].

Apart from understating data, we also need to collect user requirements like what is the goal of clustering, what relevancy measure should be considered while validating the cluster objects' similarity (or dissimilarity), etc. A good understanding of these requirements is necessary because as stated in Chapter 1, clustering algorithms are designed in such a manner that they will produce a result, no matter how the data looks or how relevant the generated clusters are. Thus, if requirements are not clear, one might never conclude whether the clusters make sense or not.

But, in an ideal setup where an engineer is deployed to perform cluster analysis or market segmentation, companies spend a lot of time and resources on the knowledge transfer processes. To reduce the efforts and time of this knowledge transfer process, we come up with a standard questionnaire - ***FEAD*** (**F**ind **E**verything **A**bout **D**ata). The aim of *FEAD* is to make one understand everything they need to know about the cluster analysis process in a few hours rather than spending weeks and months to gain the required knowledge.

This questionnaire does not only make one understand the data and process in less time but also enables a standardized practice towards cluster analysis. Imagine a situation where companies like Amazon, with billions of users, want to cluster customers as per their electronic brand preferences. The data points for such a process would be trillions in number and customer analysis no more remains a one-man job. Thus, if each engineer is provided with the same set of requirements and standardized meaning of data, everyone has clarity on how to approach the problem and what steps to take.

We came up with *FEAD* by closely analyzing the requirements of each step of the clustering process and conducting iterative interviews with a data expert to find out what knowledge is needed to perform clustering.

### 3.1.1. *FEAD* Questionnaire

*FEAD* comprises of the below mentioned questions:

1. *What does each data point represent?*
2. *What is the goal of the cluster analysis?*
3. *How do you define a relevant cluster?*
4. *On what similarity measure(s) should the cluster objects be compared?*
5. *Which are the most important data features?*
6. *How exactly does one deal with the null values (drop them, replace them with some average or general value, etc.)?*
7. *Which cases should be considered as outliers (value threshold, frequency threshold, etc.) and how does one deal with them (drop such data points, replace them with some average or general value, etc.)?*
8. *Should a specific clustering technique be adopted for cluster generation (like hierarchical clustering, partitional clustering, etc.), or should the algorithm selection be based on experimentation?*
9. *How should the clustering results be delivered (visual analysis, statistical analysis, etc.)?*

The ultimate goal of *FEAD* is to get humans acquainted with the data and clustering goal, based on which they would be able to make informed decisions in the later steps of the cluster analysis process. Thus, we propose that at the beginning of the cluster analysis, an engineer seeks answers to the *FEAD* questionnaire by setting interviews with multiple data scientists, developers, or data analysts. Rather than spending time and resources on understanding everything about the business, engineers collect answers for the above-mentioned questions, which should cover all the knowledge that one needs to know about the clustering process.

## 3.2. Adding Semantics To Categorical Features

This step is highly important because the quality of a clustering outcome is heavily dependent on extracting appropriate features. Especially for categorical data, where there exists no inherent ordering or semantic distance measure, and no numerical features which we can explore, we need to perform effective feature engineering to make sure that no data and its associated meaning is lost. Thus, one needs to derive semantics and feature engineer the data points accordingly, via a deeper understanding of the categories and their respective values.

Through this task, we introduce an approach for adding semantics to categorical data without losing information. We do so because the state-of-the-art solutions present for categorical features handling, like, encoding techniques, introduce a lot of dummy values into our dataset. Also, the motivation behind doing comes from the earlier stated challenges of

working with categorical data in Chapter 2. An example of how to do so is depicted in the tables below. Table 3.1 displays the sales entry data of a pharmacy.

| Shopper ID | Product ID | Product category | # Items Purchased |
|---|---|---|---|
| 121 | 22 | Sports | 4 |
| 121 | 21 | Household | 6 |
| 121 | 24 | Clothing | 8 |
| 121 | 25 | Sports | 9 |

Table 3.1: An example of sales entry data

From table 3.1 we can observe that 'Product ID' and 'Product category' are categorical data, thus we extract a new feature by concatenating their values, as shown in table 3.2.

| Shopper ID | Product ID | Product category | # Items Purchased | Concat |
|---|---|---|---|---|
| 121 | 22 | Sports | 4 | 22_Sports |
| 121 | 21 | Household | 6 | 21_Household |
| 121 | 24 | Clothing | 8 | 24_Clothing |
| 121 | 25 | Sports | 9 | 25_Sports |

Table 3.2: An example of extracting new feature 'Concat' from the existing categorical values

We can now drop the 'Product ID' and 'Product category' from the dataset and transform the dataset in such a manner that the values of our new extracted feature 'Concat' become the new features. The output will look like as shown in table 3.3. The output clearly shows that all the categories have semantics and values associated to them and we didn't lose any data, hence achieving the desired goal for this method.

| Shopper ID | 22_Sports | 21_Household | 24_Clothing | 25_Sports |
|---|---|---|---|---|
| 121 | 4 | 6 | 8 | 9 |

Table 3.3: An example of adding semantics to categorical features

## 3.3. Human Computational User Specific Cluster Analysis

Clustering algorithms are designed in such a way that no matter what clustering tendency does the data possess, the algorithms will produce results. Thus, depending on the nature of the target problem, an engineer needs to make a mindful analysis on whether the clustering is happening as per desired goals or not. To ensure the same, this task starts with choosing the clustering algorithms first. To do so, we re-iterate back to the answer to question 8 of the *FEAD* questionnaire. If the engineer feels that there is a specific technique that would yield the best possible clusters then they can move ahead by selecting the same, otherwise, experimentation can be performed using state-of-the-art clustering algorithms to generate clusters.

For our algorithm selection step we choose to use *KMeans* algorithm and *Agglomerative clustering* algorithm. The working of these algorithms is explained in Chapter 2. Depending on the clustering algorithm, we can choose the respective hyper-parameters. For example in the case of *KMeans*[45] - the parameters like the initial number of seeds or clusters,

or the initial cluster centroids, should be defined before one computes clusters. Table 3.4 shows the combination of different hyperparameters used to generate clusters. The meaning of each of these hyperparameters can be referred from the attached appendix A. This step generates 7 sets of 7 clusters each.

<table>
<tr><th>Algorithm</th><th colspan="3">Hyperparameters</th></tr>
<tr><td rowspan="2">KMeans</td><td>Number of Clusters</td><td colspan="2">Centroid Initialization</td></tr>
<tr><td>chosen via data scientist's insight and verified via k-elbow method</td><td colspan="2">random</td></tr>
<tr><td></td><td colspan="2">kmeans ++</td></tr>
<tr><td rowspan="5">Agglomerative</td><td>Number of Clusters</td><td>Similarity Measure</td><td>Linkage</td></tr>
<tr><td rowspan="4">chosen via data scientist's insight and verified via dendrograms</td><td rowspan="3">Euclidean</td><td>Ward</td></tr>
<tr><td>Complete</td></tr>
<tr><td>Average</td></tr>
<tr><td rowspan="2">Cosine</td><td>Complete</td></tr>
<tr><td>Average</td></tr>
</table>

Table 3.4: Algorithms and their respective hyper-parameters used to generate different sets of clusters

Once we have different cluster sets, we now try to understand whether the clustering is happening as per desired goals or not. We do so by using a data expert's insights. The main objective of this step is to choose the set of clusters that are closest to the user requirement. Here, we first present an analysis of clustering results obtained with and without applying our proposed methods on any given dataset. We then provide an overlap analysis of the different cluster sets. We believe that different algorithms should have some overlapping objects in respective clusters if the data objects possess some similarity or clustering tendency. Once the clustering tendency of the dataset is analyzed, we then present a statistical analysis and visual analysis of each cluster set. The results of the analysis are presented to the data expert using a web interface, which is designed as shown in figure 3.2.

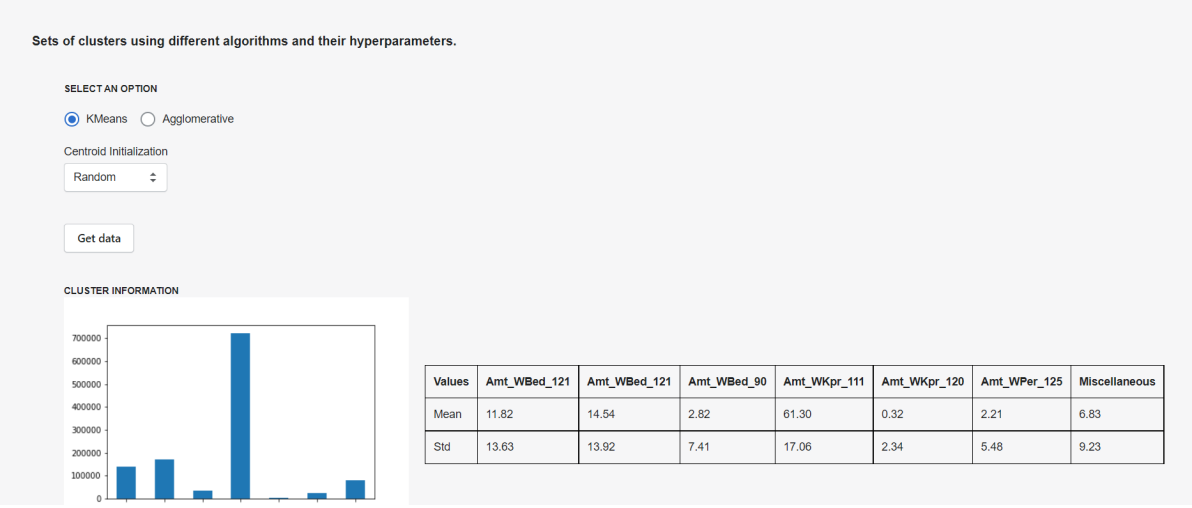



Figure 3.2: A preview of the web interface designed for a data expert to select a cluster set based on visuals and cluster statistics.

Since, in the data transformation steps, we were able to convert the categorical features into numerical features, thus it is easier to perform statistical analysis on the data set. The statistical analysis contains the collective mean and standard deviation of values of all the

numerical features in a cluster. Please note that this step would not run completely if there were categorical features in our dataset as properties like mean and standard deviation can only be applied to numerical data. An example of this is shown in table 3.5. From the results displayed in table 3.5, the data expert can infer that 'Feature 4' contributes the most to the respective cluster, as it is the one with the highest mean and frequency.

| **Property** | **Feature 1** | **Feature 2** | **Feature 3** | **Feature 4** | **Feature 5** | **Feature 6** | **Feature 7** |
|---|---|---|---|---|---|---|---|
| Frequency | 123 | 111 | 134 | **1578** | 137 | 101 | 67 |
| Mean | 10.08 | 9.03 | 6.24 | **63.99** | 11.75 | 13.07 | 12.56 |
| Standard Deviation | 7.18 | 8.92 | 20.32 | **43.45** | 13.67 | 16.13 | 18.92 |

Table 3.5: An example of data used to perform statistical analysis

The visual analysis displays the attributes (like sum, frequencies, etc.) of all the data points corresponding to respective features in a cluster. An example of this is shown in figure 3.3. From the figure, the data expert can observe that 'Feature 5' is the most popular in Cluster 1. Such observations might help them to conclude results like, as per the given information, *Cluster 1* contains data objects which have a product sales against *Feature 5*.

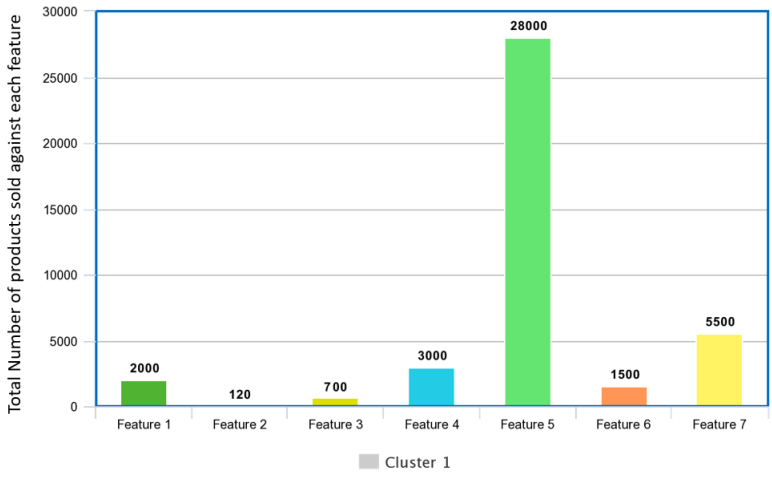



Figure 3.3: An example of data used to perform visual analysis

By the end of this step, we have a final set of clusters that matches the clustering requirement, as per the human insights. To verify these results we further perform cluster analysis using the relevancy/similarity measure. This step is elaborately explained in Chapter 4 as it is quite domain-specific. We use the output as an indication of how clusters are being formed and use humans in the next step to validate the generated clusters.

## 3.4. Human Involved Cluster Validation

As explained in Chapters 1 and 2, clustering results can not be evaluated using fixed class labels due to its unsupervised behaviors. Also, the little or no notion of ground truth makes cluster validation harder in an unsupervised setting. The literature survey also educates us on the fact that there is no universally adopted approach to choose features or clustering schemes[68] in cluster analysis.

But once we have clusters, we need a method that validates our choices of algorithms, data transformation steps, or the generated clusters in terms of relevancy. Validation gives us confidence that how well does a machine generates clusters. Thus, to validate the obtained clusters we design a human computational game - **EvalClu** (**E**valuate **C**lusters), with an educating purpose of authenticating our choices. The game aims to validate the efficiency of cluster computations by machines. The design and implementation of ***EvalClu*** is as follows:

1. *Game Goal:* To validate the machine computed clusters using human intelligence.
2. *Game Setup:*
   (a) Information (step 3) regarding two clusters is provided
   (b) Samples from the respective two clusters are provided in an excel sheet (tab: *RequiredInformation*)
   (c) For each sample (row) human provides a cluster number that they think matches with the provided information or *'none'*, in the column 'Human Label' (tab: *Let'sPlay*)
   (d) Everything about the provided data is also explained in this step. For example, what does each column or row represent?
3. *Provided Information:*
   (a) Cluster Visual: This visual provides an idea about the contribution of each feature in a particular cluster. For example, from the figure 3.3, one can easily observe that for this particular cluster values against *Feature 5* would be comparatively higher than others.
   (b) Cluster Statistics: To illustrate the mean and standard deviation of each of the given clusters against each feature, as shown in table 3.5. This provides clarity of the range of the numerical data points in a cluster.
   (c) Cluster Representatives: To illustrate examples of data points in a given cluster, as shown in table 3.6. We hypothesize that this information is very crucial for our game, as it gives actual information concerning a cluster. From our given example, it is easily observable by a human, that for a given data, the data points which have higher values against *Feature 4*, are clustered together.

| **Customer ID** | **Feature 1** | **Feature 2** | **Feature 3** | **Feature 4** | **Feature 5** | **Feature 6** | **Feature 7** | Cluster Number |
|---|---|---|---|---|---|---|---|---|
| 12345 | 11 | 10 | 01 | 1578 | 13.7 | 10.1 | 6.7 | 1 |
| 45678 | 12 | 9 | 6.24 | 2000 | 11.75 | 13.07 | 12.56 | 1 |
| 980567 | 7.18 | 8.92 | 20.32 | 2300 | 13.67 | 16.13 | 18.92 | 1 |

Table 3.6: An example of cluster representatives

4. *Inputs*:
   (a) Multiple data objects from 2 clusters, A and B.
   (b) No machine-generated cluster number provided.

5. *Steps to play*:

   (a) Humans analyze the provided information.

   (b) Humans provide a cluster number as per their understanding.

6. *Expected Output*: Human computed cluster label A, B, or none for presented data objects.

7. *Condition for Success:* Each data object has a human computed cluster label A, B, or none.

8. *Incentives:* Incentive is an important way to influence the quality of the output computed by humans. Incentives may affect many aspects of humans' behavior, including whether they do any task at all (i.e., level of participation) and how well do the humans perform each task (i.e., the accuracy and efficiency of the computation)[48]. We desire a maximum level of quality participation from humans, hence we also include incentives in our game. The best performer receives meal reimbursement of 50 euros and the first runner-up receives meal reimbursement of 30 euros. The performance is decided by the results of the accuracy percentage, explained in the next step.

Once the game is played, we then aggregate all the outputs. We now calculate the *accuracy percentage*, by dividing the total number of matched labels from the total number of data inputs. The matching percentage gives us a solid idea of cluster validation. It enables us to observe whether the cluster computation by the machine matches human intelligence. The higher the matching percentage, the higher is the machine's efficiency of generating the clusters.

The biggest contribution of ***EvalClu*** is that it introduces a novel cluster validation technique that can be adapted for any clustering application irrespective of its domain. It can be applied independently to any cluster analysis workflow, irrespective of the selections made in the prior steps. It tries to solve the problem that arises due to less notion of ground truth in an unsupervised learning setup. The user interface of ***EvalClu*** can be developed as per the application demand. We present our chosen design for *EvalClu* in appendixes C and D.

## 3.5. Summary

In this chapter, we proposed a system design that tries to solve the scientific challenges explained in section 1.2. We came up with an end-to-end human-involved cluster analysis workflow. Figure 3.4 summarises our proposed workflow in one glance.

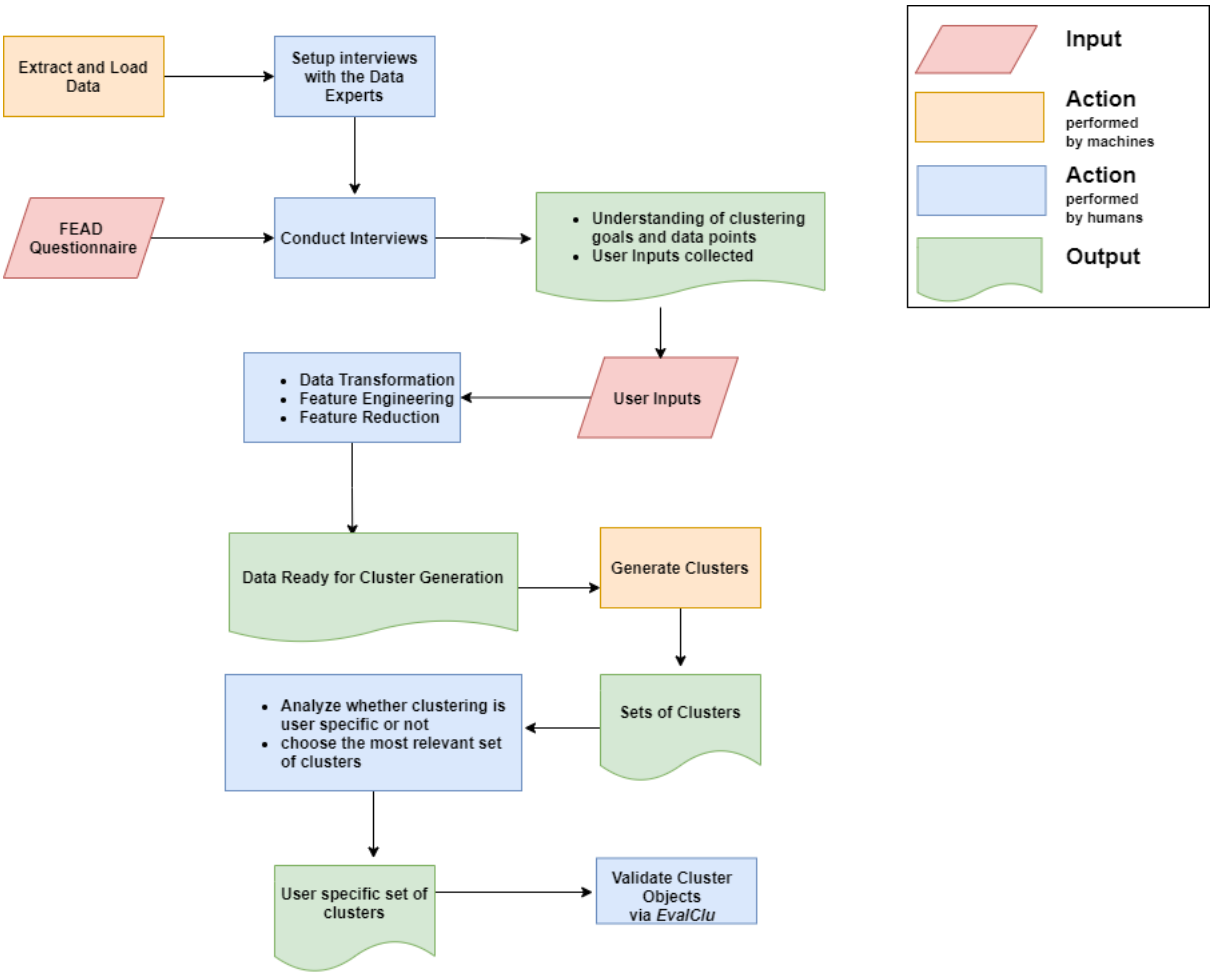



Figure 3.4: Proposed Cluster analysis workflow

# Chapter 4

# Workflow Implementation

This chapter aims to find the answer to *SQ3: How to use the proposed hybrid intelligent clustering workflow to generate customer clusters from financial data?* Throughout the chapter we discuss various experimentation steps taken to apply our cluster analysis workflow to a real time use case where we try to cluster customers based on their financial behaviours. We conclude the chapter by discussing the perceived limitations of the workflow.

## 4.1. Workflow Implementation Goal

To validate the application feasibility of our proposed workflow, we perform cluster analysis on Exact's[1] customers. The goal of this application is to cluster Exact's customers as per their financial behaviors. These behaviors are extracted from the customers' bookkeeping data.

### 4.1.1. Exact

To make this research possible, the experiments to apply our workflow to real-world data are performed using the data provided by Exact. Exact provides software for accounting, ERP, CRM, etc., to almost 350k small and medium-sized enterprises(SMEs) in the Netherlands. We focus on Exact's accounting services for this particular thesis restricting ourselves to cluster around 85k SMEs out of the lot.

### 4.1.2. Introduction To Bookkeeping

Accounting has two main parts: book-keeping and analysis. Bookkeeping is recording a company's financial transactions like buying an asset, paying electricity bills, etc. Exact's accounting software enables their customers to perform efficient bookkeeping which in return allows these customers to analyze their cash flow, knowledge about assets or to make valuable operating, investing, and financing decisions like future investments, mergers, etc.

## 4.2. Experimental Procedure

The experimental procedure is divided into two parts.

[1] `https://www.exact.com/`

### 4.2.1. Experiments - Generating clusters via our proposed methodologies

This is performed as per the steps defined in chapter 3. The experiments for the same are conducted in an online setup due to Covid-19 restrictions. Each of the steps taken to complete the cluster generation part is defined elaborately in the following subsections of this chapter.

1. **Introduction to dataset:** Table 4.1 shows 3 samples from the raw dataset. Each row represents a transaction of a particular Division in the given time using multiple features. The meaning of each of the data points is explained in the next subsection.

| Division | Date | RGS | Transaction_Type | Debtor | General_Account_Type | Transaction_Amount |
|---|---|---|---|---|---|---|
| 1652033 | 2018-07-01 | WOmz | 40 | False | 110 | -47949.859253 |
| 1825640 | 2019-09-23 | WBed | 40 | True | 121 | 12045.820022 |
| 1119288 | 2019-01-15 | WKpr | 40 | True | 120 | 0.30000 |

Table 4.1: Samples from use-case dataset

2. **Gathering user requirements and understanding data:** In this step, we gather all the information needed to perform cluster analysis, via *FEAD* questionnaire. To make this happen, we conducted 3 interviews with Exact's Data experts who understood data and were acquainted with clustering goals. Following are the answers collected via the aforementioned interviews.

   (a) Each of the values in table 4.1 means the following:

   - *Division* - a unique number that defines one entity or business or company that uses Exact's services. It is a numerical data type. The extracted raw data contains approximately 200k *Divisions*.
   - *Year* - the year in which the transaction happened, this can be further segregated into the quarter, months, or days. It is a numerical datatype. For this particular use case, we use the data from 2018 and 2019 only.
   - *RGS* - called Referentie GrootboekSchema in Dutch, is a reference classification that contains all the ledgers which are required to report to the Dutch government and most of the ledgers used for internal reporting[2]. To further understand, an example of an RGS value is 'WKpr' which refers to transactions against *'Purchase Value Sales'*. It is a categorical data type. In our dataset we have 13 *RGSs* values.
   - *Transaction_Type* - it represents the nature of the transaction, i.e. was the transaction made a purchase entry or bank payment, etc. These values are internally generated by Exact following the bookkeeping standards. It is a categorical data type. In the given dataset we have three *Transaction_Type* values.
   - *Debtor* - This is a boolean data type that describes whether a transaction was a debit or credit. If the 'Debtor' value is false, it means that the transaction type is a credit entry and debit entry otherwise.

[2]https://referentiegrootboekschema.nl/english

- *General_Account_Type* - This is a categorical data type that represents the general ledger account type. "A general ledger account is a record in which is recorded a specific type of transaction. A separate general ledger account is set aside for each specific type of transaction. For example, within the general area of inventory assets, there may be separate general ledger accounts for raw materials inventory, work-in-process inventory, finished goods inventory, and merchandise (purchased) inventory"[3]. The values against General_Account_Type in our dataset are internally generated by Exact following the bookkeeping standards. In the given dataset we have 24 different values of *General_Account_Type*.
- *Transaction_Amount* - a numerical data type that represents the amount of each transaction in euros. The negative sign in this feature represents that the entry is a credit entry and debit otherwise. In the given dataset the values of *Transaction_Amount* ranges from -99999997952 to 1282429249.

(b) The goal of cluster analysis is to group '*Divisions*' based on their financial behaviors. Financial behavior is quite a subjective term, but as per our goal, it can be defined as the properties associated with a particular transaction. For example, the transaction is made in which year, for how much amount and booked via which combination of *RGS*, *Transaction_Type* and *General_Account_Type*. Through the clustering process, we want to be able to categorize the '*Division*' according to their behavior for commercial and analytical reasons.

(c) A relevant cluster can be defined as a group of *'Divisions'* with similar transaction behaviors. For example '*Divisions*' who book most of their transactions pertaining to a similar combination of *RGS* and *General_Account_Type*, should be in one cluster.

(d) Combination of *RGS*, *Transaction_Type* and *General_Account_Type* can be taken into account for comparing the similarity between data objects of a cluster.

(e) The most important data features are *RGS*, *Transaction_Type* and *General_Account_Type* as they form the base of a transaction.

(f) To handle the null values, one should know that as per the Exact software used for bookkeeping only *Transaction_Amount* can have a null value. One can drop the transactions with *Transaction_Amount* equal to 0. As null amount doesn't give any information about a financial transaction.

(g) There can be outliers in *Transaction_Amount* due to human errors. We can drop all the transactions of such '*Divisions*', as for now we want to perform clustering on the cleanest '*Divisions*'.

(h) No specific clustering technique is preferred. It should be decided via experimentation.

(i) The results should be delivered in form of text results (tables), cluster visuals, and statistical analysis of numerical attributes. It would also be nice if some analysis can showcase the clustering tendency of the dataset.

[3]https://www.accountingtools.com/articles/what-is-a-general-ledger-account.html

3. **Baseline implementation:** Before implementing the further steps of our proposed workflow, we first start with a baseline implementation following the standard pipeline of cluster analysis. In this implementation we do not involve humans at any stage and perform the following steps:

   (a) Clean the data, drop null values, remove outliers using the IQR method with 25th and 75th quartile.

   (b) Standardise the numerical values using MinMaxScaler.

   (c) Convert categorical features into numerical features using state-of-the-art encoding techniques.

   (d) Find the number of clusters using the K-Elbow method and perform clustering using the KMeans algorithm.

   Figure 4.1 shows the distribution of objects in each cluster.

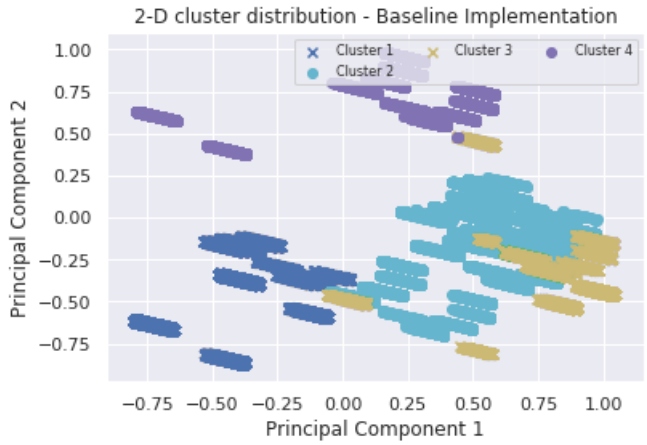



Figure 4.1: Distribution of data objects within clusters in baseline implementation

We can observe that 4 clusters are being generated. The clusters are formed based on financial transactions and do not meet the requirement of clustering the *'Divisions'*. Also, the state-of-the-art encoding techniques introduce 89% data scarcity. Thus, from the baseline implementation, we conclude that we need further elaboration in each of the cluster analysis steps and the standard clustering techniques do not meet our desired goals. Hence, we move on by applying the methodologies defined in our proposed workflow.

4. **Data Transformation And Feature Engineering:** To complete this task, we perform the following steps:

   - Convert *Transaction_Amount* from float to rounded off integer, to make data cleaner.
   - Transform dataset as per the negative and positive values of *Transaction_Amount*, with the help of domain expert's knowledge. This step eliminates the negative sign from the dataset.

- We then eliminate the *Divisions* which have less than 20 entries in the two years time frame.
- The range of *Transaction_Amount* indicates that there are outliers in the data, as per the data expert. Thus, we eliminate *Divisions* with outlier in *Transaction_Amount* using the IQR method as defined below:

  In the IQR method, the dataset is divided into quartiles[65]. This method is applied as below:

  - Find the first quartile, Q1. The first quartile point indicates that the given percentage of the data points are below that value.
  - Find the third quartile, Q3. The first quartile point indicates that the given percentage of the data points is above that value.
  - We now calculate IQR as follows:

    $$IQR = Q3 - Q1$$

  - Define the lower limit as:

    $$LowerLimit = Q1 - n * IQR$$

    where $n$ is chosen as per the user requirements.
  - Define the upper limit as:

    $$UpperLimit = Q3 + n * IQR$$

    where $n$ is chosen as per the user requirements.

  All the data points lying outside the lower limit and upper limit range, are treated as outliers.

  The first quartile is at 10% and the third quartile is at 90%. Also the '$n$' is chosen as 30.
- Next we choose the most contributing *RGS*, *Transaction_Type* and *General_Account_Type* based on their frequencies amongst the transactions. This leads to 7 *RGS* values, 7 *General_Account_Type* and 1 *Transaction_Type* value.
- Next we drop the columns '*Year*', '*Debtor*' and *Transaction_Type* as the dataset is already transformed handling these values, thus they are not needed for further transformation. Table 4.2 shows how the dataset looks after this step.

| Division | RGS | General_Account_Type | Transaction_Amount |
|---|---|---|---|
| 10000080 | WBed | 111 | 9500 |
| 10000080 | WBed | 111 | 9870 |
| 1825640 | WBed | 121 | 897 |
| 1119288 | WKpr | 120 | 58900 |

Table 4.2: Samples from dataset after Data Transformation Task

- Moving ahead we now group the dataset using *Division*, *RGS* and *General_Account_Type*. This is needed because one *Division* might have multiple entries, as shown in table 4.2. Thus, we need to club them together, because, in the end, we need one row representing all the details concerning one *Division*. As we want to cluster *Divisions* and not their transactions. Table 4.3 shows the result of this step. As compared to the values in table 4.2, we can observe that the transactions associated with reoccurring *Division* are now clubbed together.

| Division | RGS | General_Account_Type | Aggregated_Transaction_Amount |
|---|---|---|---|
| 10000080 | WBed | 111 | 19370 |
| 1825640 | WBed | 121 | 897 |
| 1119288 | WKpr | 120 | 58900 |

Table 4.3: Samples from dataset after groupby operation

- We now add semantics to the categorical features. We concatenate *RGS* and *General_Account_Type* values for each row and pivot the table with respect to *Division*. As a result, the concatenated combination of *RGS* and *General_Account_Type* become new features and the amount corresponding to these combinations become the data values. By the end of this step, we have now added semantics to the categorical attributes and transformed the entire dataset into a numerical type.
- Before moving to the next step, we first have to standardize the dataset values. Generally, machine learning algorithms work better if they receive a normalized dataset[59]. Thus we normalise the *Aggregated_Transaction_Amount* per *Division* (in our case per row) using the following formula:

$$Amount = (Amount / Total Amount of a Division) * 100$$

The insight of deriving this formula comes from data expert's intuitions. As if we normalize the *Transaction_Amount* within a range as per the state-of-the-art solutions like MinMaxScaler[4], the data will lose the business essence. As is the value of one *Transaction_Amount* might impact other. We want to cluster according to the *Aggregated_Transaction_Amount* distribution. Thus, we want a unique scale across *Division* to be able to compare them to one another. In this scenario, taking the rate of the aggregated amount per event is guarantying linear transformation and interpretability.

For example, let's say there are two customers, one which is an IT company and the other which is a hairdresser. Naturally, the *Transaction_Amount* of the former will be greater than the latter, throughout the dataset. If we normalize *Transaction_Amount* based on this, every small business will be impacted by the bigger business, irrespective of the economy of that particular sector. Hence, this step is crucial to maintain the original information of each *Division*.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

- From the concatenation step in adding semantics task we have 49 features (combinations) in the engineered dataset as there exist 7 *RGS* values and 7 *General_Account_Type* values. The major goal of this step is to have the most important features in the dataset and eliminate those which do not contribute much and introduces data scarcity. We do so by using the visual analysis (figure 4.2) of each of the features and decide a threshold of feature contribution.

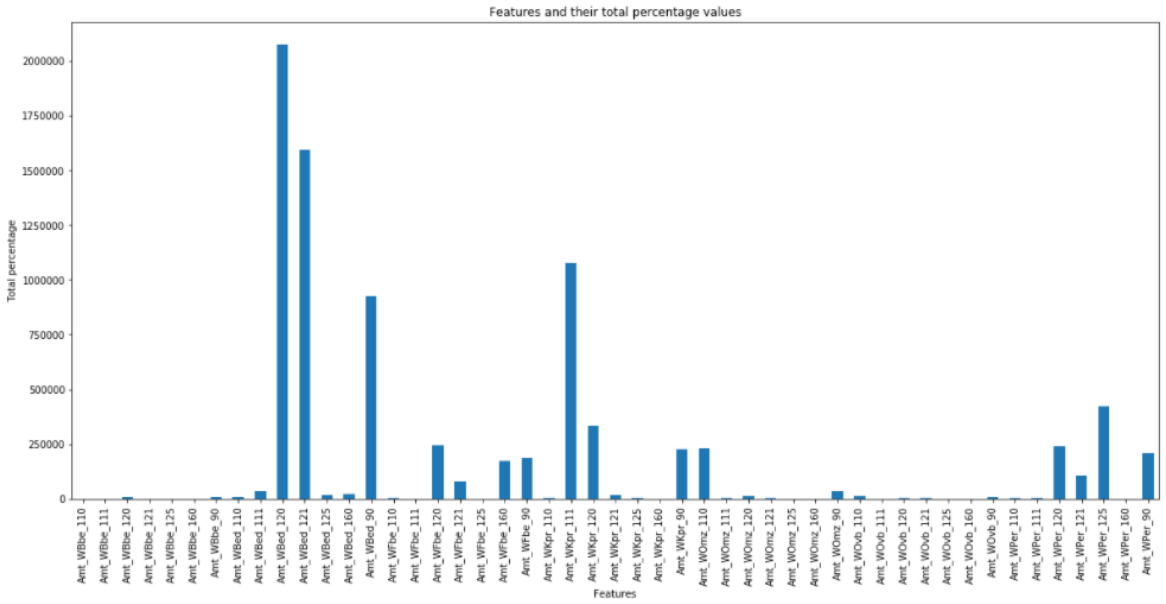



Figure 4.2: Visual analysis of features and their respective values before Feature Reduction Task

From figure 4.2, we could visualize that not all features have significant values, thus we set the threshold at 250000. Any feature that has a value above 250000, is kept originally in the dataset and the remaining are reduced to one feature called "Miscellaneous". By doing so, we reduce data scarcity and make sure that no data is lost. The results of this step can be seen in figure 4.3.

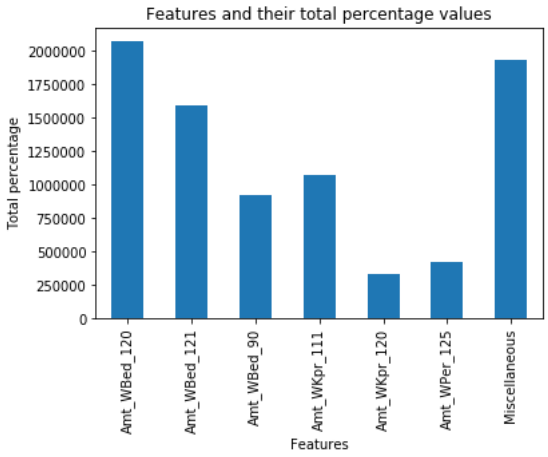



Figure 4.3: Visual analysis of the reduced features and their respective values

This step could however be improved by other ways which are more discriminant in nature like tf-idf or advanced analytics. But due to time constraints, we

limited ourselves to go for the threshold values/contributions of features.

By the end of this step, we now have our dataset ready for the cluster generation step. Data samples from this engineered dataset are presented in table 4.4.

| Division | Amt_WBed_120 | Amt_WBed_121 | Amt_WBed_90 | Amt_WKpr_111 | Amt_WKpr_120 | Amt_WPer_125 | Miscellaneous |
|---|---|---|---|---|---|---|---|
| 929771 | 84 | 3 | 2 | 1 | 3 | 0 | 7 |
| 1062815 | 3 | 0 | 55 | 2 | 9 | 8 | 23 |
| 919055 | 70 | 3 | 1 | 2 | 1 | 1 | 22 |

Table 4.4: Samples from dataset ready for cluster generation

5. **Cluster Generation:** Before generating clusters, first we need an idea of the number of clusters required. We collect this input from the data expert. Based on the engineered data set, the data expert makes an educated guess that the number of clusters required would be 7 which is equal to the number of features, hypothesizing that each cluster will be distinct from others, at least in respect to one dominating feature. We generate clusters using algorithms and hyperparameters stated in table 3.4.

   - **Algorithm - Kmeans**

     (a) Verifying the number of cluster using K-elbow method (figure 4.4).

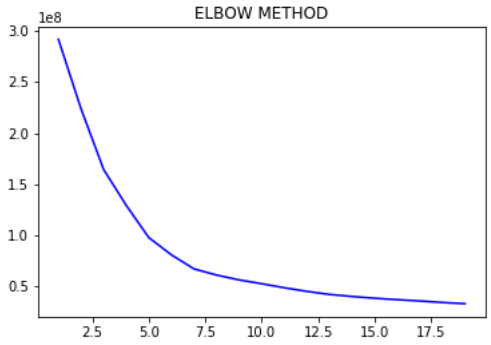



Figure 4.4: Verifying the number of cluster input by data expert which validates the 7 number choice, using K-Elbow method

     (b) Generating clusters using KMeans and its different hyperparameters (figure B.1).

   - **Algorithm - Agglomerative Hierarchical Clustering**

     (a) Verifying the number of cluster using dendrograms (figure 4.5). This can be verified by the number of different colours present in the dendrogram.

     (b) Generating clusters using agglomerative hierarchical clustering algorithm and its different hyperparameters (figure B.2).

6. **Human Computational User Specific Cluster Analysis:**

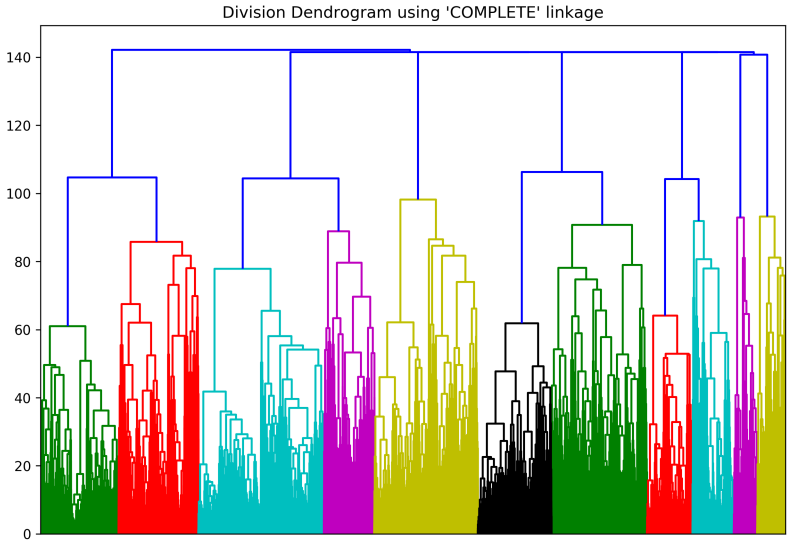
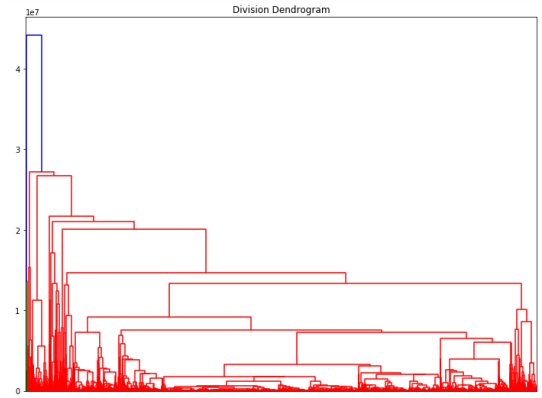
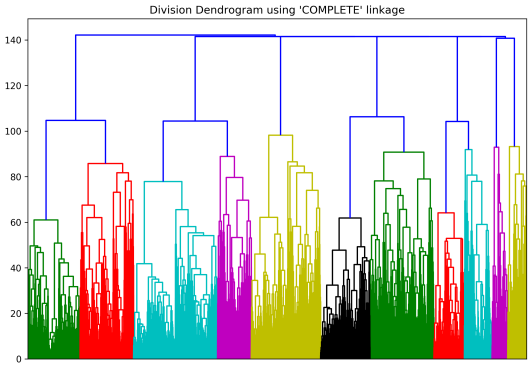



Figure 4.5: Verifying the number of cluster input by data expert which validates the 7 number choice, using dendrograms

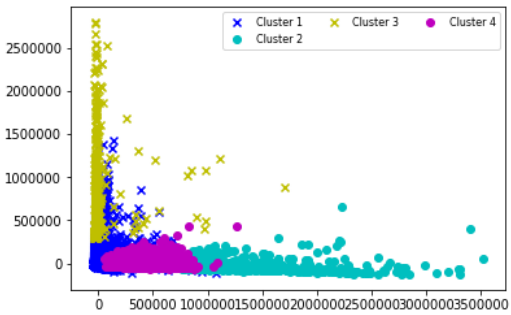
- In this step, we first investigate the before and after results obtained after applying the data transformation and feature engineering steps. Figure 4.6 depicts this comparison. From the figure, we can observe that without our proposed methodologies, mostly all the *Divisions* are grouped as shown in divisions dendrogram ***a***. This observation itself is enough to provide us confidence in our data transformation and feature engineering step.
- Next, we perform a cluster overlapping analysis to verify the clustering tendency of the dataset. We do so by first providing an overlap analysis of the different cluster sets as shown in table 4.5. The object overlapping is calculated between all possible combinations of the cluster sets created using KMeans and Agglomerative algorithms. We believe that these results give us an understanding of the clustering tendency of the given dataset. The highlighted results in table 4.5 represent that there exists an object overlap between different pairs of clusters which means that a significant number of objects falls in the same cluster, irrespective of which algorithm we use. We believe that this confirms the clustering tendency of the dataset.
- We now present the visual (examples attached in appendix B.3) and statistical results of each cluster generated using different hyperparameters to a data expert. These results are presented using the web interface (figure 3.2) developed specifically for carrying out this task. After a thorough analysis, the data expert decides the best set of clusters are obtained by using an agglomerative hierarchical clustering algorithm (complete linkage and cosine similarity measure). This decision is based on the following inputs:
  - Clarity of mean and standard deviation of each feature of clusters within a set.
  - similarity measure as our dataset contains data scarcity, and as per data expert's knowledge in such cases, cosine distance works better than Euclidean distance.



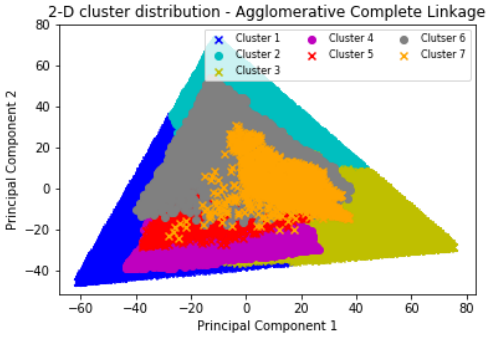

(a) Division Dendrogram before the Data Transformation and Feature Engineering




(b) Division Dendrogram after the Data Transformation and Feature Engineering




(c) Clustering before the Data Transformation and Feature Engineering




(d) Clustering after the Data Transformation and Feature Engineering

Figure 4.6: Comparison of the before and after implementation of Data Transformation and Feature Engineering

| | **Agglomerative Clusters** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Cluster #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **KMeans Clusters** | 1 | 0.18 | 1.43 | **38.29** | 0.00 | 0.61 | 0.00 | 0.01 |
| | 2 | 0.06 | 0.61 | 3.17 | **37.60** | 0.30 | 0.93 | 0.02 |
| | 3 | 0.31 | 0.80 | 0.23 | 0.01 | 0.18 | **37.62** | 0.06 |
| | 4 | 0.90 | 1.05 | 0.04 | 0.29 | **41.26** | 0.00 | 0.00 |
| | 5 | **40.52** | 0.22 | 0.66 | 0.88 | 0.00 | 0.00 | 0.00 |
| | 6 | 0.19 | 0.07 | 0.34 | 3.41 | 0.43 | 0.00 | **25.41** |
| | 7 | 1.26 | **41.07** | 0.03 | 0.19 | 0.03 | 0.00 | 0.00 |

Table 4.5: Overlapping analysis between cluster objects generated using Kmeans and Agglomerative algorithm

  - 2-D representation of the clusters which indicates object overlapping within a set of clusters (attached in appendixes B.1 and B.2). However, such visuals do not give a very concrete idea around clustering because we convert multiple features into 2 using Principal Component Analysis to obtain these visuals.

- Once the data expert has chosen the most relevant set of clusters, we perform

a detailed (dis)similarity analysis on the clusters' objects. Before doing so we changed the features name into a more human-readable manner, i.e. renaming them to what does each *RGS* or *General_Account_Type* value means. Table 4.6 show the clusters' (dis)similar properties.

| Cluster Number | (dis)similar property |
|---|---|
| 0 | Cluster dominated by Divisions with higher expenses booked against 'Purchase Value Sales' RGS through 'Costs of Goods' General Ledger Account |
| 1 | Cluster dominated by Divisions with higher expenses booked against 'Expenses' RGS and 'Sales' General Ledger Account |
| 2 | Cluster dominated by Divisions with higher expenses booked against a variety of RGS and General Ledger Account, excluding the ones mentioned in other clusters |
| 3 | Cluster dominated by Divisions with higher expenses booked against 'Expenses' RGS and 'Other Costs' General Ledger Account |
| 4 | Cluster dominated by Divisions with higher expenses booked against 'Expenses' RGS and 'General' General Ledger Account |
| 5 | Cluster dominated by Divisions with higher expenses booked against 'Personnel' RGS and 'Employee Cost' General Ledger Account |
| 6 | Cluster dominated by Divisions with higher expenses booked against 'Purchase Value Sales' RGS through 'Other Costs' General Ledger Account |

Table 4.6: (dis)similarity analysis of the final set of clusters

### 4.2.2. Experiments - Validating clusters via our proposed methodologies

Through the previous experimentation step, we now have our final set of clusters. In this step, we now validate the cluster objects as per the EvalClu design, stated in chapter 3. We ask humans to generate the cluster labels for various data objects based on the provided information. These experiments are performed by setting up EvalClu for 11 employees at Exact. These users are designated data experts, data engineers, or data analysts. The experiments are conducted online due to Covid-19 restrictions. Each participant is provided with the game setup tutorial, attached in appendix C. The data points which need to be labeled into different clusters are provided in shared excel sheets (due to the privacy policy of Exact). We provide 200 data points to each of the employees because of time and human resource constraints. However, our game is designed in a manner that can be scaled up to any number of human resource usage. An input/output interface prototype is attached in appendix D. At the end of the experiments, the outputs are also delivered by the participants through online applications (evaluation form, excel sheet, etc.). However, due to lack of human resources, we could only verify this for 10% of the data samples. We conclude this step by collecting human-generated cluster labels and match them with the machine-generated labels. This calculation provides our results for one of our later defined evaluation metrics: *Cluster Validation Accuracy.*

## 4.3. Summary

In this chapter, we presented the workflow implementation steps on Exact's data. We apply the proposed novel methodologies on Exact's customers' bookkeeping data to cluster them as per their financial behaviors. We gathered the required information using *FEAD* questionnaire and then transformed data by creating semantics in categorical features. Once, we generated clusters using KMeans and agglomerative algorithms, we then performed a user-specific cluster analysis to confirm whether the generated clusters meet user demands or not. Our analysis results show that the different cluster objects do inhibit (dis)similar financial behaviors. In the end, we validated the cluster objects using *EvalClu*. The results of which are discussed in Chapter 5. During experimentation we also perceived a few limitations which are as follows:

- Data Scarcity: We proposed a method that adds semantics to the categorical features and attempts to replace the entire dataset into numerical values. This works well for our use case. However, there are chances that this method introduces null values if the number of categorical features increases. Because the method increases the dimensions/features of the dataset and thus, introduces more data sparsity. But again this is subjective to the type of dataset that we deal with.

- Missing Global Perspective: In our proposed cluster validation game (*EvalClu*), we present information associated with only two clusters, to each human. And if a human thinks, that the data points presented belong to none of these, they can mark the human-generated label as "None". However, we feel that if the information about all clusters is presented, it may yield a better understanding of the entire clustering, rather than just the two respective clusters. This might yield a better matching percentage.

# Chapter 5

# Evaluation and Results

This chapter presents the evaluation of multiple human-involved steps in our proposed hybrid intelligent clustering model. We discuss the results against different evaluation metrics defined in this chapter, based on the collected user data. We conclude the chapter by identifying the limitations of experiments.

## 5.1. Model Evaluation Metrics

The proposed end to end human computational workflow will be evaluated by using the following evaluation metrics:

### 5.1.1. Cluster Validation Accuracy

As established in chapter 2, there is no standard way of validating the clusters and their respective objects. Thus, for measuring the relevancy of clustering results concerning the clustering goal, we compute the matching percentage of machine-generated cluster labels and human-generated cluster labels. We assume here that humans are always right and treat the human-generated cluster labels as the ground truth. Thus, the higher the matching percentage, the more relevant cluster generation has been. Also, a higher matching percentage indicates how well the machine has computed clusters. We collect the human-generated labels via our proposed human computational game i.e *EvalClu*.

### 5.1.2. Execution Time

In our model, we used human intervention at the following stages:

- FEAD questionnaire for collecting required information about the clustering process
- Selecting the most relevant set of clusters using the designed web interface
- Validating cluster objects via EvalClu

Thus, for the time metric, we ask users to mention the time (in minutes) needed by humans for each of the above-mentioned tasks.

### 5.1.3. Cognitive Task Load

The task load is computed using the task load index proposed by Hart et al.,[35]. This metric aims to find the cognitive demands of *EvalClu* in terms of:

- Mental demand i.e. how much thinking was required to provide the cluster number to the data objects.
- Physical demand i.e. how much physical effort was required to accomplish the task (was the task restful or laborious).
- Temporal demand i.e. was there any time pressure that was felt to complete the task.
- Performance i.e. how well do you think you performed?
- Effort i.es how hard did you work to accomplish the task.
- Frustration i.e. how frustrated, irritated, or agitated (or calm, composed, or satisfied) you were while completing the task.

The participants are asked to provide an answer to questions shown in figure 5.1 ranging from 0 to 10 (with 1 scale increment, where 0 means low and 10 means high). The task load index (TLX) per criteria is calculated by adding the scores provided by all the users and dividing the sum by the number of users. ("RAW" TLX[35]). A lower TLX score indicates a lower perceived cognitive load.


Figure 5.1: Cognitive Load Evaluation based on NASA-TLX

### 5.1.4. User Engagement

User engagement (UE) and its measurement is a factor of great interest for human computer interaction workflows[55]. The more the engagement is, the more users are motivated and inclined towards finishing the task with high quality. We evaluate the user engagement with the help of the ***User Engagement Survey*** (UES) proposed by O'Brien et al.,[55]. The user engagement is evaluated for two components of our model i.e. the web interface which enables a data expert to select the most relevant set of clusters and *EvalClu*.

Participants answer the following questions as shown in figures 5.2 and 5.3 on a scale from 1 to 5, where 1 indicates strongly disagree and 5 indicates strongly agree. The score per criteria is calculated by adding the scores provided by all the users and dividing the sum by the number of users.

## 5.2. Evaluation Result

In this section, we discuss the obtained results while evaluating the model using the above-defined metrics.

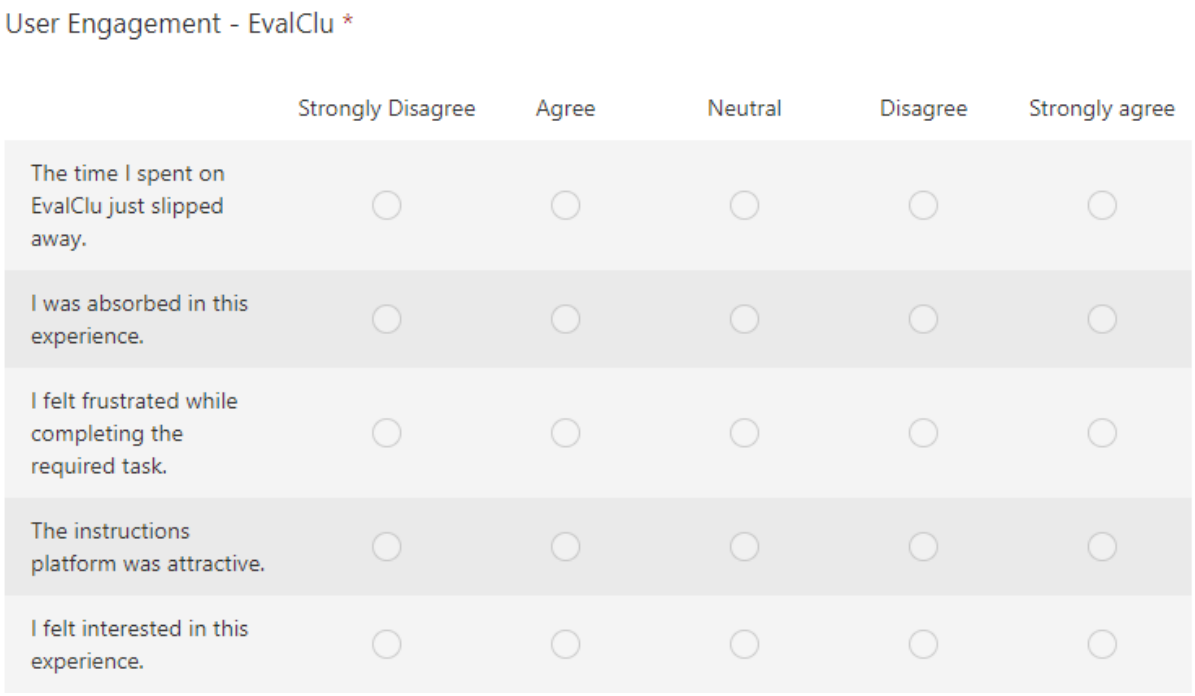



Figure 5.2: User Engagement evaluation of EvalClu

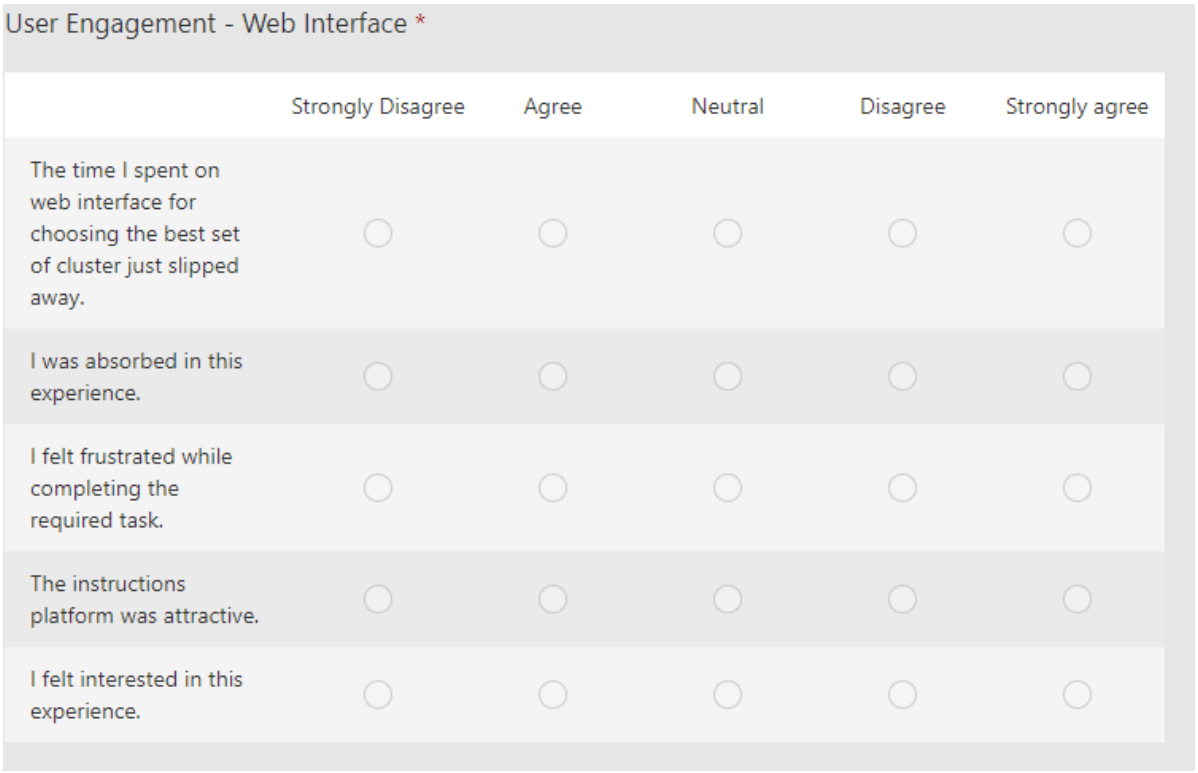



Figure 5.3: User Engagement evaluation of Web Interface

### 5.2.1. Cluster Validation Accuracy

The figure 5.4 displays the matching percentage achieved by the machine against the cluster labels provided by each Exact employee. The matching percentage results show that on

an average **87%** of the machine-generated cluster labels matched human-generated cluster labels. This means that out of 2200 samples, machines were able to match the human cluster labels for 1914 samples. This clearly explains that machines can spot the difference between similar and dissimilar objects in terms of human perception. This accuracy validates our choices for each of the prior steps i.e. data transformation, feature engineering, algorithm selection, and cluster analysis results.

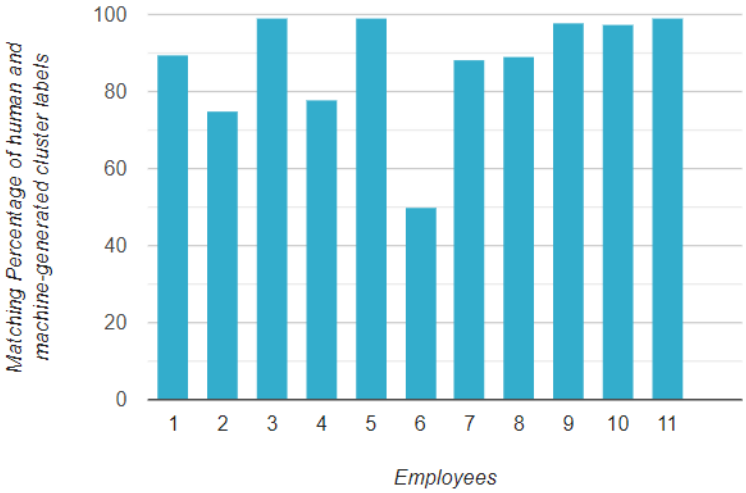



Figure 5.4: Matching percentage achieved by Exact's Employees

### 5.2.2. Execution Time

The table 5.1 displays the average time taken by humans to complete the above-mentioned tasks. The average time is taken to collect answers for *FEAD* questionnaire, i.e. 42 minutes indicates that developing a set of clustering-related questions brought down the knowledge transfer efforts to minutes instead of days and weeks.

The time is taken to choose the best set of clusters from 7 sets, where each set contains 7 clusters, thus a data expert ends up analyzing 49 visuals and statistical data tables respectively. The data expert takes 120 minutes to choose the most relevant clusters, which means 1.22 minutes for a visual or a table. This indicates that the presented data (via Web Interface) was not too complex and systematically structured, which enabled the data expert to make quick decisions.

| Task | Average Time (in minutes) | Number of humans involved |
|---|---|---|
| Interview - FEAD Questionnaire | 42 ±10 | 3 |
| Web Interface (User specific cluster analysis) | 120 ±0 | 1 |
| EvalClu | 18 ±13.68 | 11 |

Table 5.1: The average time taken by humans to complete various tasks

*The EvalClu* game is played successfully by a human for an average of 18 minutes for 200 samples. This indicates that it was quite easy for humans to decide the objects' labels. This gives us confidence that the cluster objects are easily distinguishable.

### 5.2.3. Cognitive Task Load

Figure 5.5 displays the results against task load for each criterion. From the results, we can see that in terms of demands, all the three criteria mental, physical, and temporal demand have been scored low with an average of 4.09, 3.90, and 3.54 respectively. Which indicates that the task was not at all demanding. The average score of performance which is 5.63 shows that the users were somewhat sure about their generated labels. On a scale of 0 to 10, the average score of 4.27 against efforts shows that decent efforts were required to complete the task, which again validates that the presented information (including the data points) helped the employees to generate required outputs effortlessly. Lower scores of frustration indicate that the users were calm, composed, or satisfied while completing the task. Overall, the lower scores for each of the criteria indicate that the EvalClu tasks do not require much effort from users.

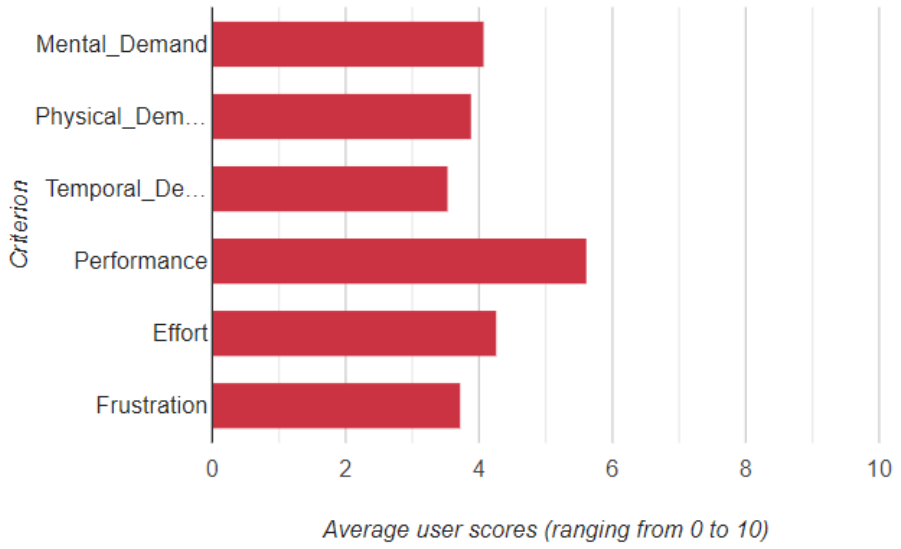



Figure 5.5: Average task load scores per criterion

### 5.2.4. User Engagement

The table 5.2 states the average score given by users for different user engagement criteria. For EvalClu the scores were given by 11 Exact's employees who participated in the cluster validation game. From the achieved scores we can conclude that the users were interested in completing the task and were seldom frustrated. The Web Interface was although used by one user only, thus scores don't include a variety of opinions, still, the platform received decent scores. We can conclude that the data expert was engaged in the task and enjoyed using the platform although they needed to process a lot of information to produce the required output.

| Evaluation Criteria (Scored on a range of 1 to 5) | Average Score EvalClu | Average Score Web Interface |
|---|---|---|
| The time spent on platform just slipped away | 3.45 ±0.8 | 3 ±0.0 |
| I was absorbed in this experience | 3.85 ±0.5 | 4.12 ±0.0 |
| I felt frustrated while completing the task | 2.1 ±1.2 | 1 ±0.0 |
| The platform was attractive | 4.2 ±0.1 | 3 ±0.0 |
| I felt interested in this experience | 4.2 ±0.2 | 4.1 ±0.0 |

Table 5.2: The average user engagement score for EvalClu and Web Interface

## 5.3. Discussions

In this section, we discuss the implications that can be made from the above-obtained results. We also discuss the limitations of the performed experiments.

### 5.3.1. Implications

The experiments show that our human-involved end-to-end cluster analysis workflow can generate user-specific clusters using Exact's data. The objects within each cluster share some of the other financial behavior. Starting from the initial steps, the intermediate results like the comparison stated in figure 4.6, give us confidence in our proposed method for adding semantics to the categorical data. The other three proposed methodologies i.e. *FEAD* questionnaire, human involved user-specific cluster analysis, and the human computational cluster validation game - *EvalClu* have been evaluated using well-defined mentioned metrics. Hence, we further state the implications around these methods based on the collected evaluation data and the discussions with data experts at Exact post experimentation. The achieved cluster validation accuracy shows us that the machine achieved an accuracy of more than 70% against the 10 sets of human-generated labels. However, for one individual set, only 49.8% of machine-generated cluster labels matched the human-generated labels. Thus, we analyzed these particular human-generated labels and found out that the quality was missing from the human end as they spent only 4.32 minutes labeling the 200 samples, whereas the other humans spent around 18 minutes each to label the 200 samples. Hence, from this, we can conclude that the average execution time spent by the humans to generate labels, on one hand, assures quality but on the other hand taking only 5.4 seconds on average to generate cluster label for each data object indicate that the humans were easily able to spot the (dis)similarity amongst different cluster objects.

The average 42 minutes spent on collecting answers for *FEAD* questionnaire show that each question was elaborately discussed during the interviews. The data experts felt that the formulated questions covered the collection of entire cluster analysis requirements. This data proves that we were able to achieve the purpose of the standardized approach of collecting user requirements in less time as compared to the time spent on knowledge transfers in an ideal organizational setup. The average time is taken by the data expert to choose the most relevant set of clusters also shows that it was less complicated than anticipated. As the web interface displayed cluster information of 7 sets with 7 clusters in each set. The data shows that on average the data expert spent 17.14 minutes on each set of clusters to visualize and understand the statistics.

The cognitive task load scores of EvalClu show that the task of labeling 200 data ob-

jects was not demanding. The average mental demand, physical demand, and temporal demand scores being 4.09, 3.90, and 3.54 respectively out of 10 show that the task was moderately challenging i.e. not too easy and not too hard. During our interaction with the data expert, we found that they were confident about their performance, and most of them had to put minimal effort to complete the game. However, two of the data experts felt extremely frustrated due to the online setup of the game and working from home. They assured us that the game setup had nothing to do with the frustration and it was external environmental conditions that caused the agitation.

From the user engagement scores, one can deduce that the employees at Exact liked both the user interfaces, i.e. the EvalClu and Web Interface for cluster analysis. They were absorbed in the experience and seldom felt frustrated. During our discussions with the data experts, we found out that the EvalClu interface was accurately designed with all the required information needed to play the game. This satisfaction indeed aligns well with the received scores. For the web interface, the data experts felt that it could have been more attractive, however, they were fine with the fact that the main purpose was being served and it was easier to follow.

### 5.3.2. Limitations

The biggest limitation of the performed experimental steps is that the number of resources (time and participants) for the human computational game was quite limited, due to which we could obtain only 2200 human-generated cluster labels. Although the machine was able to generate 87% average accuracy for the cluster objects, yet we feel that, these results could have been more convincing if we could run the experiment on a larger sample.

Also, we believe that the experiments' online setup due to Covid-19 regulations made it a bit hard for us to closely examine the results' implications, especially in terms of the cognitive load because it is possible that working from home might frustrate some employees more than others and that might have impacted the overall human performance.

# Chapter 6

# Conclusion

In this thesis, we tried to solve the scientific challenges that arise while performing cluster analysis. One of the challenges is that generally, state-of-the-art clustering algorithms are developed in a broader sense without targeting any specific applications. Further, they are used in multiple application domains. However, because these algorithms lack domain-specific information and user-specific input, they do not always produce relevant results. Also, datasets with categorical features are harder to cluster as such features lack semantics. Another challenge is that, unlike classification tasks, clustering cannot be evaluated using well defined labels and is quite subjective to the need of users. Also, the little notion of ground truth makes cluster validation harder in an unsupervised setting.

To solve the stated challenges we came up with an end-to-end cluster analysis workflow. We introduced 4 novel methods including a method to add semantics to categorical features, a novel human-involved cluster validation game and a human computational user specific cluster analysis method. To reach our solution, we divided our research into three sub-questions.

To find answers to the first sub-question i.e. *What are the state of the art in data clustering and hybrid intelligence approaches?*, we performed an exhaustive literature survey to find an intersection between clustering and hybrid intelligence. We identified various research works and novel contributions that validate our hypothesis that hybrid intelligence is used in various applications to solve the clustering challenges. Motivated by this to answer our second question i.e. *How and where to use hybrid intelligence in cluster analysis workflow?* We built a model that involves humans in almost every stage of cluster analysis. This model does not only generate user-specific relevant clusters but also incorporates a novel cluster validation game.

Once the model was built we then tried to answer the third sub-question i.e. *How to use the proposed hybrid intelligent clustering workflow to generate customer clusters from financial data?* We did so by applying our model to cluster Exact's customers based on their bookkeeping (financial) data. By the end of our model application on Exact's data we were able to generate clusters relevant to Exact's requirements and were also able to validate the same using intelligence and cluster understanding of Exact's employees.

Last but not the least, through this research work we are now able to answer our main research question i.e.*How can we use hybrid intelligence in cluster analysis workflow to generate user-specific clusters and evaluate them?* To justify this we can say that human intelligence can be combined with every step of cluster analysis to generate and validate user-specific clusters. Human intelligence can be used to understand data, transform and

feature engineer data for clustering algorithms, understand the clustering process itself, or validate the generated clusters.

## 6.1. Future Work

Due to time constraints, we could not implement and explore all the ideas and methodologies. We believe that the thesis (proposed hybrid intelligent cluster analysis model) can be extended and further improved. Firstly, we can use humans in the loop to improve the clustering algorithm. This can be achieved by using EvalClu results as feedback to modify the working of the algorithms. To do so one can analyze the non-matching human-generated and machine-generated cluster labels.

Secondly, we believe that creating a global perspective might increase the matching accuracy of human-generated and machine-generated cluster labels. In the EvalClu, we propose to present only two clusters' information to a human. However, we feel that if global knowledge of cluster generation is needed for an application, the entire clusters information can be provided to a human who validates clusters using EvalClu. This, however, might increase the task's cognitive load which can be further investigated by experimentation.

Thirdly we believe that feature selection techniques can be further enhanced. In our application, we choose features based on their numerical contributions. However, we believe that this can be further improved by using discriminative methods like TF-IDF or advanced analytics methods like Pearson Correlation.

# Appendix A

# Hyperparameters and their meanings

In this appendix we explain the meaning of the hyperparameters of the clustering algorithms used to generate our clusters.

- Euclidean distance similarity: It is a default measure of similarity for many clustering algorithms. It is calculated using the following formula[58]:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

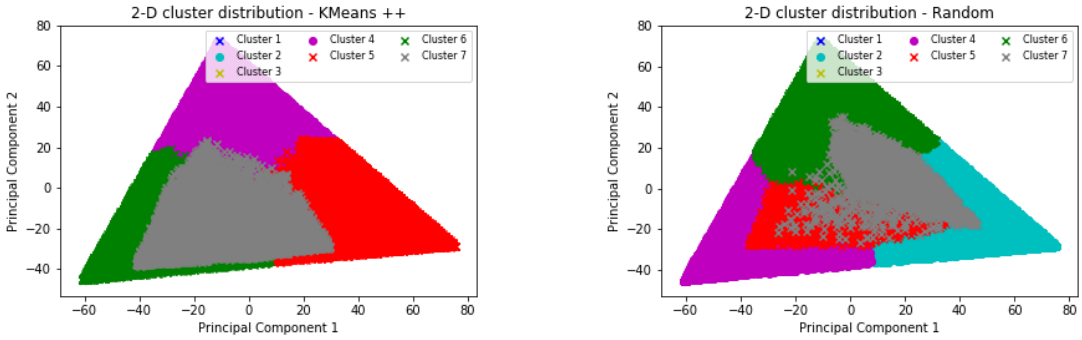
- Cosine similarity: Cosine similarity measures whether two vectors point the same direction or not[37]. It is calculated using the blow formula:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2}$$

- K-Means centroid initialization:
  - K-Means 'Random' centroid initialization: "In this approach k random data points are selected from the dataset and used as the initial centroids"[49].
  - K-Means 'Kmeans++' centroid initialization: "In this approach, the first centroid is a randomly selected data point, and then the the next centroids are calculated from the remaining data points based on a probability proportional to the squared distance away from a given point's nearest existing centroid"[49].
- Agglomerative algorithm's linkages:
  - Complete Linkage: "Complete-linkage is where distance is measured between the farthest pair of data points in two clusters"[70].
  - Average Linkage: "Average-linkage is where the distance between each pair of data points in each cluster are added up and divided by the number of pairs to get an average inter-cluster distance"[70].
  - Ward Linkage: "Ward linkage is where the distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster. Th next data point is chosen to minimize the increase in ESS at each step"[57].

# Appendix B

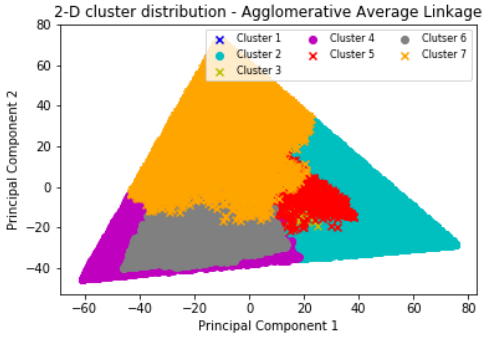
# Experiment Results



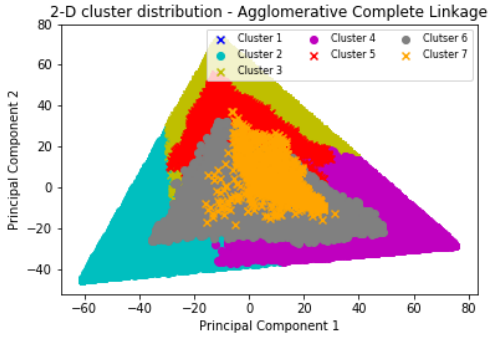

(a) KMeans++ centroid initialization, Similarity Measure: Euclidean distance

(b) Random centroid initialization, Similarity Measure: Euclidean distance

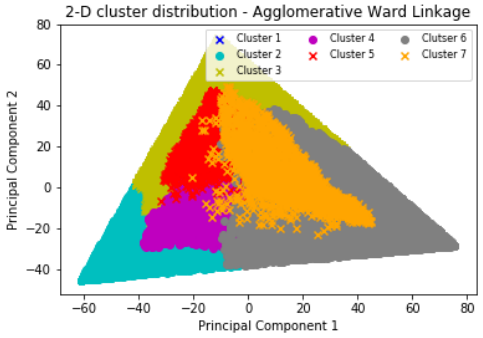
Figure B.1: Generating clusters using Kmeans algorithm and its different hyperparameters



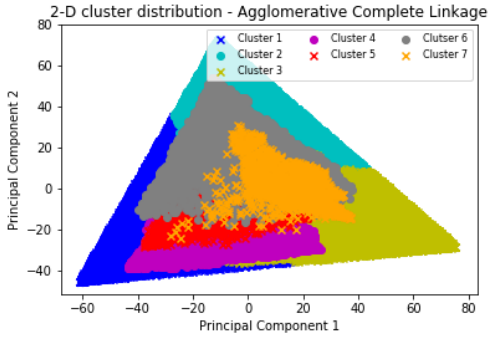

(a) Linkage: Average, Similarity Measure: Cosine distance



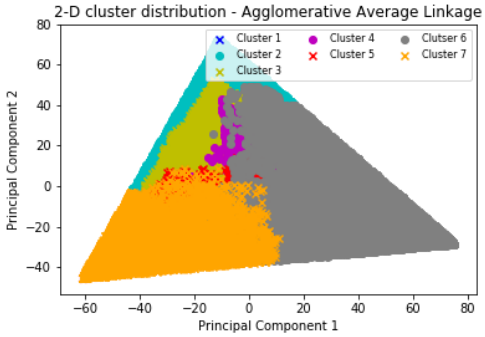

(b) Linkage: Complete, Similarity Measure: Cosine distance




(c) Linkage: Ward, Similarity Measure: Euclidean distance




(d) Linkage: Complete, Similarity Measure: Euclidean distance




(e) Linkage: Average, Similarity Measure: Euclidean distance

Figure B.2: Generating clusters using agglomerative hierarchical clustering algorithm and its different hyperparameters

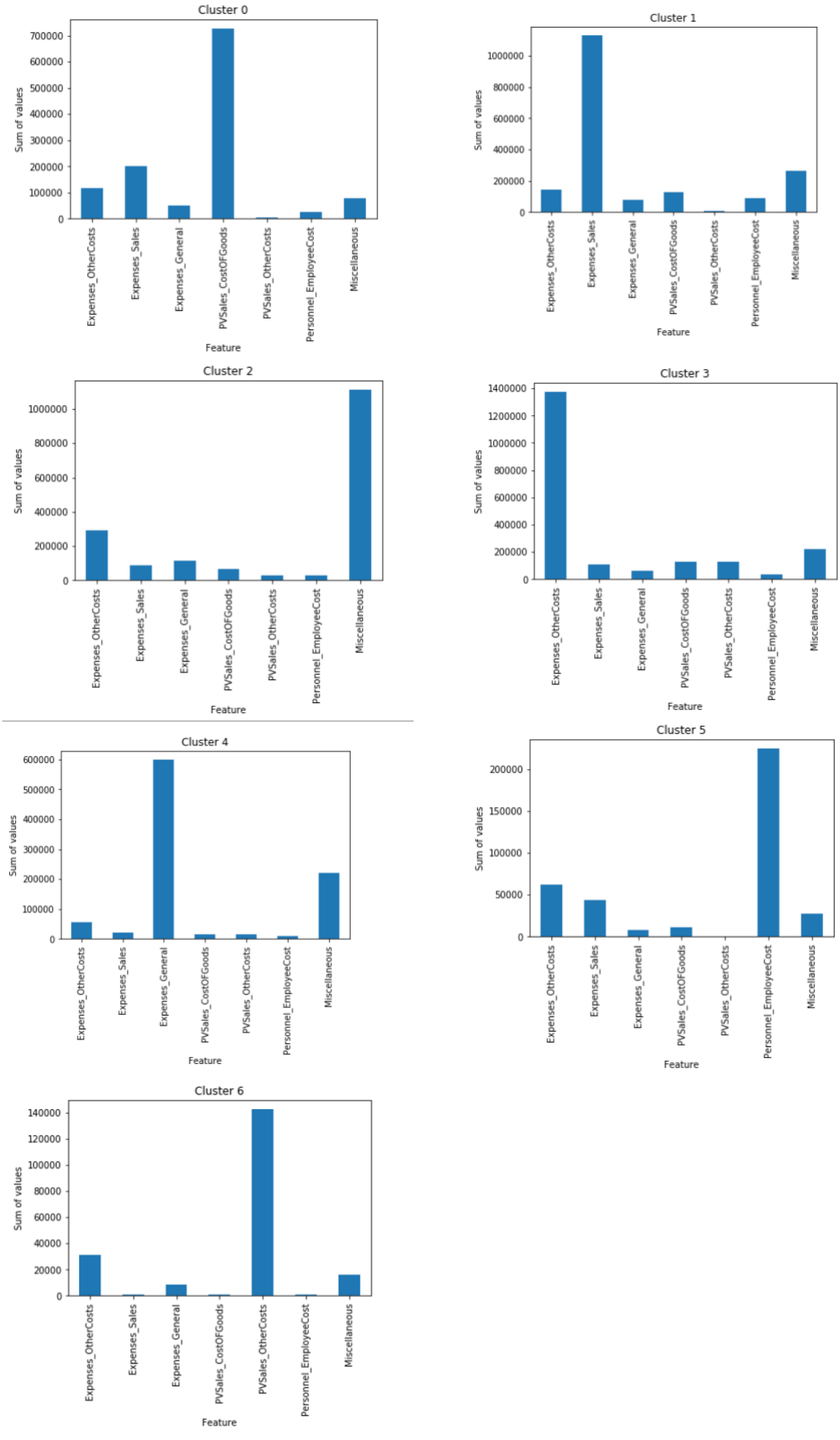



Figure B.3: Final Clusters and the distribution of total amount against each of their feature

# Appendix C

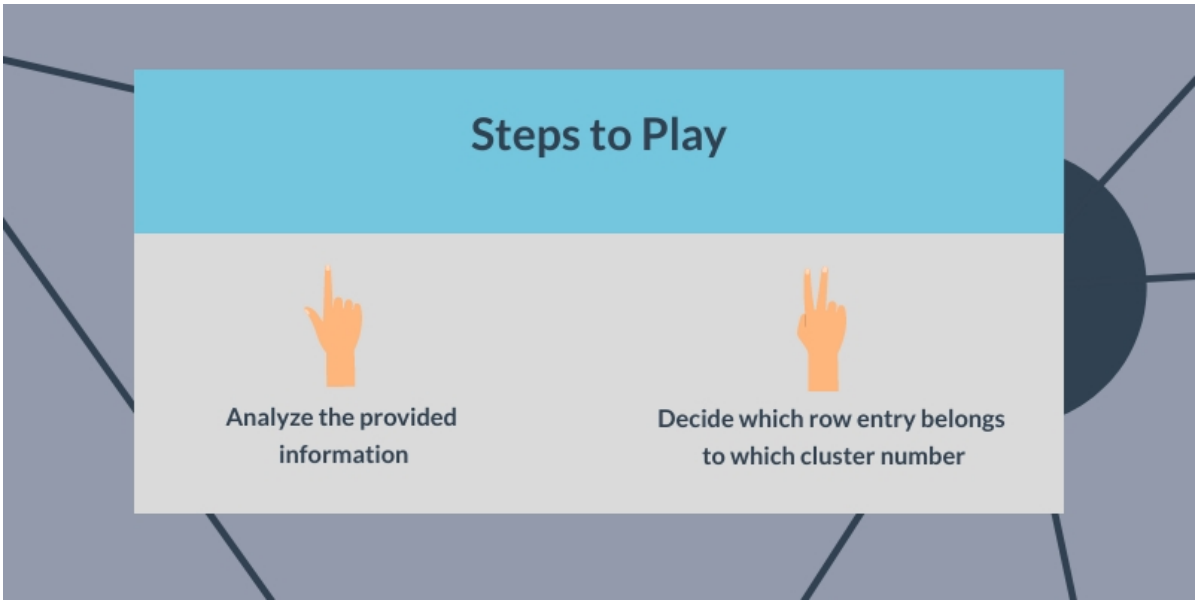
# EvalClu - Game Setup

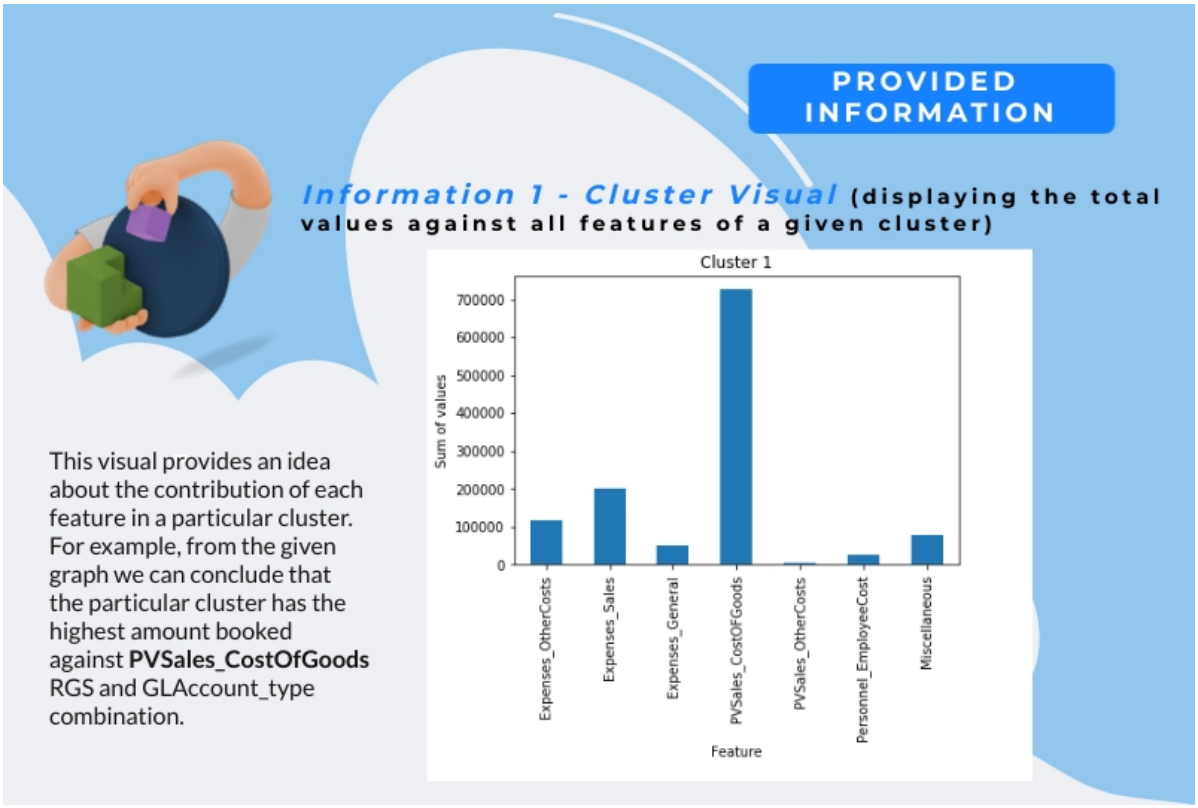
# Instructions!!

- General information (as demonstrated later) regarding two clusters is provided
- Samples from the respective two clusters are provided in an attached excel sheet (tab - RequiredInformation)
- For each sample (row) provide a cluster number that you think matches with the provided information or none, in the column 'Human Label' (tab - Let'sPlay)
- Please keep a note of time spent on this task, we need it for completing the evaluation form

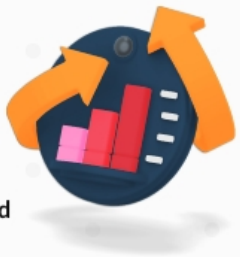
# More about data

- The data set contain 8 columns and multiple rows. The first column displays the *'Division' number.*
- The remaining columns represent the total amount (normalized percentage) against an RGS and GLAccount_type. For example, column '*Expenses_ OtherCosts*' represents "Amount booked against RGS *WBed which refers to Expenses,* and general account type *121 which refers to 'Other Costs' account*".
- The column '*Miscellaneous'* represents the amount against a mix of multiple RGSs and General Account types, that were not very significant contributors in terms of booking amount.
- Each row displays the amount booked by the respective 'Division' during 2018 and 2019 per RGS and General Account type.


Steps to Play
Analyze the provided information
Decide which row entry belongs to which cluster number


PROVIDED INFORMATION
Information 1 - Cluster Visual (displaying the total values against all features of a given cluster)
This visual provides an idea about the contribution of each feature in a particular cluster. For example, from the given graph we can conclude that the particular cluster has the highest amount booked against PVSales_CostOfGoods RGS and GLAccount_type combination.
Cluster 1
Sum of values
0
100000
200000
300000
400000
500000
600000
700000
Expenses_OtherCosts
Expenses_Sales
Expenses_General
PVSales_CostOfGoods
PVSales_OtherCosts
Personnel_EmployeeCost
Miscellaneous
Feature

## *Information 2 - Cluster Statistics* (Illustrating the mean and standard deviation of each of the given cluster against each feature)



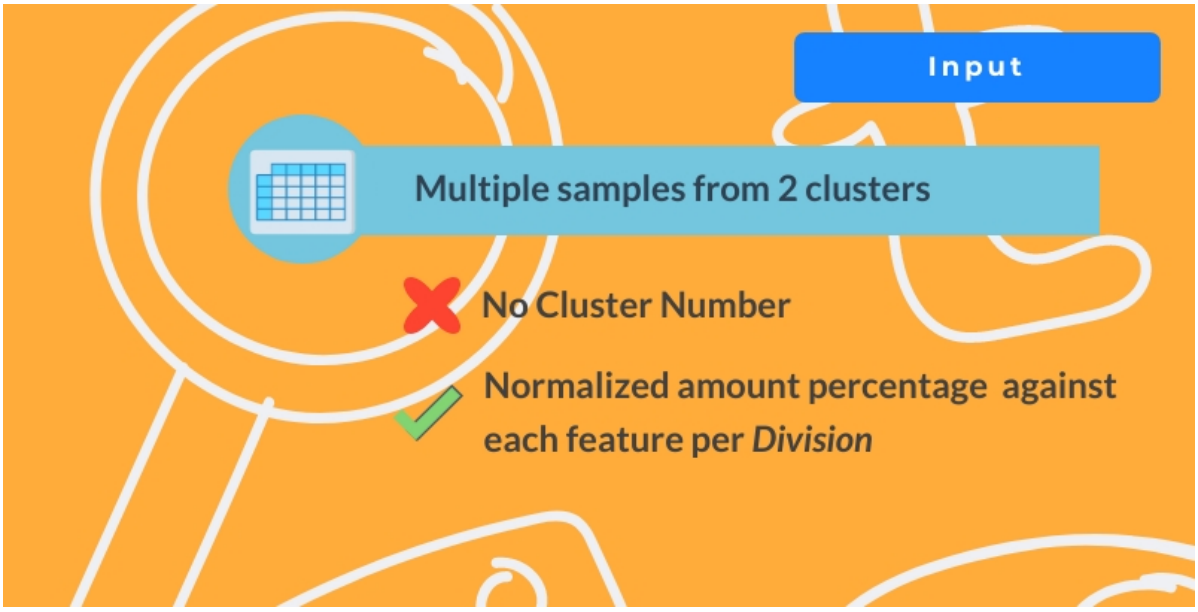
- The statistics presented below inform one about the range of normalized amounts against each feature.
- For example, the table below indicates that for the given cluster the highest amount will be mostly associated with 'PVSales_CostOfGoods' and the other features have comparatively lower amounts booked against them.

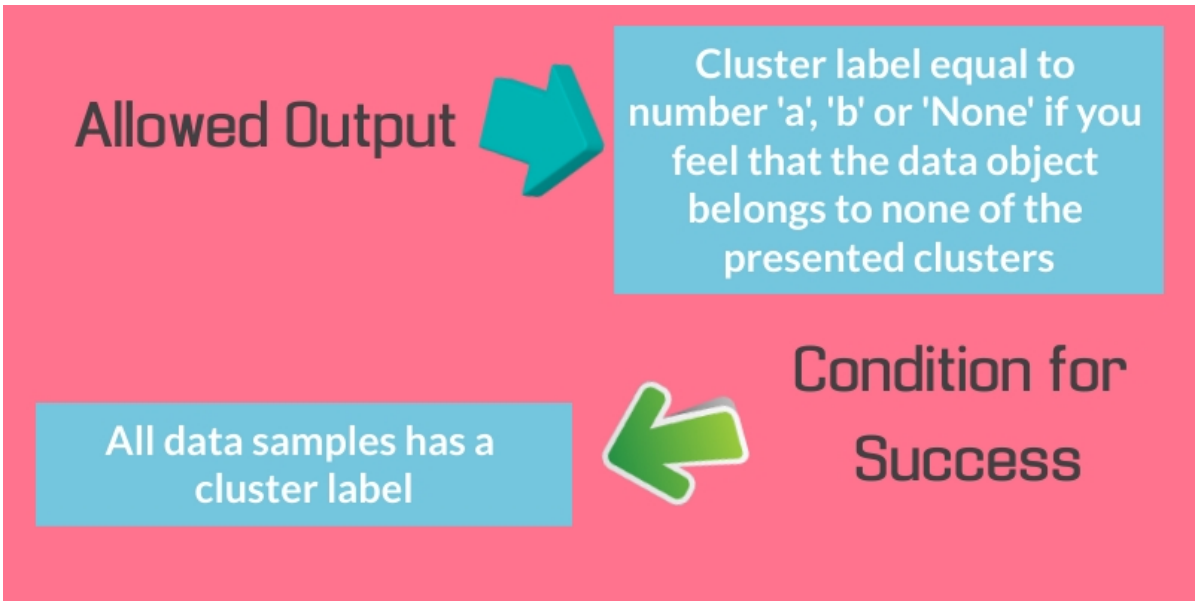
| Values | Expenses_OtherCosts | Expenses_Sales | Expenses_General | PVSales_CostOFGoods | PVSales_OtherCosts | Personnel_EmployeeCost | Miscellaneous |
|---|---|---|---|---|---|---|---|
| Mean | 9.767434 | 16.724767 | 4.279855 | 59.890732 | 0.540315 | 2.263679 | 6.420401 |
| Std | 12.059103 | 15.655542 | 10.986704 | 18.323699 | 4.041032 | 6.132927 | 8.610019 |

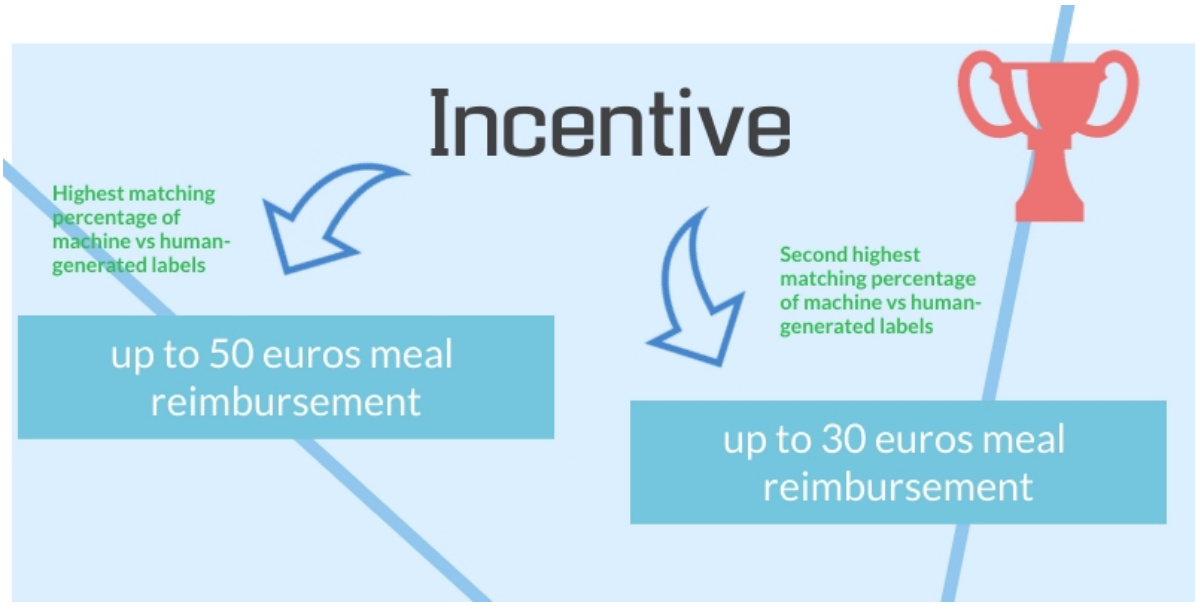
## *Information 3 - Cluster Representatives* (Illustrating examples of data points in a given respective cluster)

| Division | Expenses_OtherCosts | Expenses_Sales | Expenses_General | PVSales_CostOFGoods | PVSales_OtherCosts | Personnel_EmployeeCost | Miscellaneous | Machine Label |
|---|---|---|---|---|---|---|---|---|
| 1626049 | 9 | 5 | 0 | 78 | 0 | 4 | 5 | 0 |
| 2196261 | 0 | 43 | 1 | 53 | 0 | 0 | 3 | 0 |
| 1918276 | 26 | 0 | 0 | 73 | 0 | 0 | 2 | 0 |

- This information provides examples of data points which are associated with the given cluster.

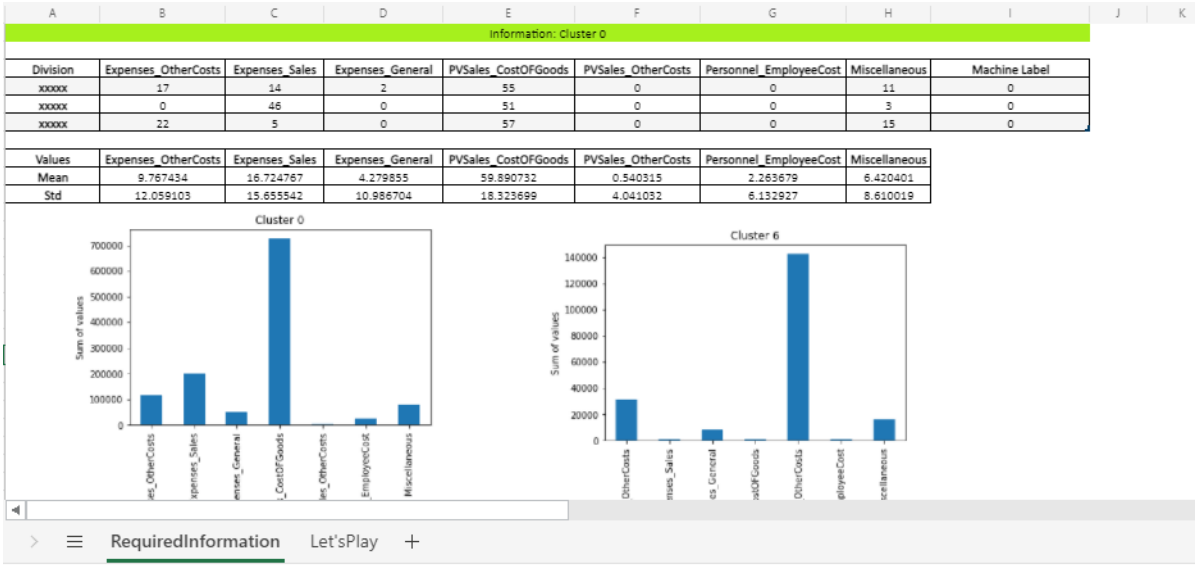

Input
Multiple samples from 2 clusters
No Cluster Number
Normalized amount percentage against each feature per *Division*


Allowed Output
Cluster label equal to number 'a', 'b' or 'None' if you feel that the data object belongs to none of the presented clusters
Condition for Success
All data samples has a cluster label


Incentive
Highest matching percentage of machine vs human-generated labels
up to 50 euros meal reimbursement
Second highest matching percentage of machine vs human-generated labels
up to 30 euros meal reimbursement

# Appendix D

# EvalClu - Input/Output Interface



Information: Cluster 0

| Division | Expenses_OtherCosts | Expenses_Sales | Expenses_General | PVSales_CostOFGoods | PVSales_OtherCosts | Personnel_EmployeeCost | Miscellaneous | Machine Label |
|---|---|---|---|---|---|---|---|---|
| xxxxx | 17 | 14 | 2 | 55 | 0 | 0 | 11 | 0 |
| xxxxx | 0 | 46 | 0 | 51 | 0 | 0 | 3 | 0 |
| xxxxx | 22 | 5 | 0 | 57 | 0 | 0 | 15 | 0 |

| Values | Expenses_OtherCosts | Expenses_Sales | Expenses_General | PVSales_CostOFGoods | PVSales_OtherCosts | Personnel_EmployeeCost | Miscellaneous |
|---|---|---|---|---|---|---|---|
| Mean | 9.767434 | 16.724767 | 4.279855 | 59.890732 | 0.540315 | 2.263679 | 6.420401 |
| Std | 12.059103 | 15.655542 | 10.986704 | 18.323699 | 4.041032 | 6.132927 | 8.610019 |



| Let's Play | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Division | Expenses_Other Costs | Expenses_ Sales | Expenses_ General | PVSales_Cost OFGoods | PVSales_Other Costs | Personnel_Employee Cost | Miscellaneous | Human Label |
| xxxx | 1 | 0 | 2 | 97 | 0 | 0 | 0 | |
| xxxx | 0 | 16 | 58 | 22 | 0 | 0 | 4 | |
| xxxx | 46 | 1 | 0 | 53 | 0 | 0 | 0 | |
| xxxx | 13 | 0 | 0 | 76 | 3 | 0 | 8 | |
| xxxx | 0 | 50 | 0 | 42 | 0 | 0 | 8 | |
| xxxx | 20 | 0 | 0 | 75 | 0 | 3 | 1 | |
| xxxx | 8 | 31 | 0 | 23 | 0 | 0 | 38 | |
| xxxx | 4 | 9 | 2 | 86 | 0 | 0 | 0 | |

# Bibliography


[1] Michael R Anderberg. *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*, volume 19. Academic press, 2014.

[2] Periklis Andritsos, Panayiotis Tsaparas, C Sammut, and GI Webb. Categorical data clustering., 2010.

[3] Juhee Bae, Tove Helldin, Maria Riveiro, Sławomir Nowaczyk, Mohamed-Rafik Bouguelia, and Göran Falkman. Interactive clustering: a comprehensive review. *ACM Computing Surveys (CSUR)*, 53(1):1–39, 2020.

[4] Jonathan Baron, Barbara A Mellers, Philip E Tetlock, Eric Stone, and Lyle H Ungar. Two reasons to make aggregated probability forecasts more extreme. *Decision Analysis*, 11 (2):133–145, 2014.

[5] Sumit Basu, Danyel Fisher, Steven Drucker, and Hao Lu. Assisting users with clustering tasks by combining metric learning and classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[6] Anant Bhardwaj, Juho Kim, Steven Dow, David Karger, Sam Madden, Rob Miller, and Haoqi Zhang. Attendee-sourcing: Exploring the design space of community-informed conference scheduling. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, 2014.

[7] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[8] Lydia Boudjeloud-Assala, Philippe Pinheiro, Alexandre Blansché, Thomas Tamisier, and Benoît Otjacques. Interactive and iterative visual clustering. *Information Visualization*, 15(3):181–197, 2016.

[9] Jonathan Bragg, Daniel Weld, et al. Crowdsourcing multi-label classification for taxonomy creation. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 1, 2013.

[10] Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, 2011.

[11] Nan Cao, David Gotz, Jimeng Sun, and Huamin Qu. Dicon: Interactive visual analysis of multidimensional clusters. *IEEE transactions on visualization and computer graphics*, 17(12):2581–2590, 2011.

[12] José A Castellanos-Garzón, Carlos Armando García, Paulo Novais, and Fernando Díaz. A visual analytics framework for cluster analysis of dna microarray data. *Expert Systems with Applications*, 40(2):758–774, 2013.

[13] M Emre Celebi. *Partitional clustering algorithms*. Springer, 2014.

[14] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Neural information processing systems*, volume 22, pages 288–296. Citeseer, 2009.

[15] Joseph Chee Chang, Aniket Kittur, and Nathan Hahn. Alloy: Clustering with crowds and computation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3180–3191, 2016.

[16] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2334–2346, 2017.

[17] Shuo Chang, Peng Dai, Lichan Hong, Cheng Sheng, Tianjiao Zhang, and Ed H Chi. Appgrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 348–358, 2016.

[18] Yixin Chen and Li Tu. Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2007.

[19] Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992–2001, 2013.

[20] Jason Chuang and Daniel J Hsu. Human-centered interactive clustering for data analysis. In *Conference on Neural Information Processing Systems (NIPS). Workshop on Human-Propelled Machine Learning*, 2014.

[21] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.

[22] Vladimir Dobrynin, David Patterson, Mykola Galushka, and Niall Rooney. Sophia: an interactive cluster-based retrieval system for the ohsumed collection. *IEEE Transactions on Information Technology in Biomedicine*, 9(2):256–265, 2005.

[23] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[24] Avinava Dubey, Indrajit Bhattacharya, and Shantanu Godbole. A cluster-level semi-supervision model for interactive clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 409–424. Springer, 2010.

[25] Mary T Dzindolet, Linda G Pierce, Hall P Beck, and Lloyd A Dawe. The perceived utility of human and automated aids in a visual detection task. *Human Factors*, 44(1):79–94, 2002.

[26] B Everitt, Sabine Landau, M. Leese, and Daniel Stahl. *Cluster Analysis*. 01 2011. ISBN 9780470749913. doi: 10.1002/9780470977811.ch8.

[27] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[28] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. Cactus—clustering categorical data using summaries. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83, 1999.

[29] Daniel G Goldstein, R Preston McAfee, and Siddharth Suri. The cost of annoying ads. In *Proceedings of the 22nd international conference on World Wide Web*, pages 459–470, 2013.

[30] Ryan Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. Neural Information Processing Systems, 2012.

[31] Geoffrey J Gordon. Approximate solutions to markov decision processes. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1999.

[32] S Guha, R Rastogi, and K Shim. Rock: Robust clustering using links. In *Proceedings of the International Conference on Data Engineering ICDE*, volume 99, 1999.

[33] Steffen Hadlak, Heidrun Schumann, Clemens H Cap, and Till Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2267–2276, 2013.

[34] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1):191–215, 1997.

[35] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[36] Hannes Heikinheimo and Antti Ukkonen. The crowd-median algorithm. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 1, 2013.

[37] Anna Huang et al. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56, 2008.

[38] Zhexue Huang and Michael K Ng. A note on k-modes clustering. *Journal of Classification*, 20(2):257–261, 2003.

[39] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[40] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[41] Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, 344:68–125, 1990.

[42] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52, 2011.

[43] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[44] Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. Crowdsourcing in computer vision. *arXiv preprint arXiv:1611.02145*, 2016.

[45] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

[46] Chinmay E Kulkarni, Richard Socher, Michael S Bernstein, and Scott R Klemmer. Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 99–108, 2014.

[47] Ludmila I Kuncheva and Lakhmi C Jain. Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern recognition letters*, 20(11-13):1149–1156, 1999.

[48] Edith Law and Luis von Ahn. Human computation. *Synthesis lectures on artificial intelligence and machine learning*, 5(3):1–121, 2011.

[49] Boyang Li et al. An experiment of k-means initialization strategies on handwritten digits dataset. *Intelligent Information Management*, 10(02):43, 2018.

[50] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[51] T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.

[52] Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. Human-powered sorts and joins. *arXiv preprint arXiv:1109.6881*, 2011.

[53] Arpita Nagpal, Arnan Jatain, and Deepti Gaur. Review based on data clustering algorithms. In *2013 IEEE conference on information & communication technologies*, pages 298–303. IEEE, 2013.

[54] Iftekhar Naim, Daniel Gildea, Walter Lasecki, and Jeffrey P Bigham. Text alignment for real-time crowd captioning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 201–210, 2013.

[55] Heather L O'Brien, Paul Cairns, and Mark Hall. A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. *International Journal of Human-Computer Studies*, 112:28–39, 2018.

[56] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

[57] Shweta Sharma, Neha Batra, et al. Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 568–573. IEEE, 2019.

[58] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

[59] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[60] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011.

[61] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

[62] W Fred Van Raaij. *Understanding consumer financial behavior: Money management in an age of financial illiteracy*. Springer, 2016.

[63] Jennifer Wortman Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *J. Mach. Learn. Res.*, 18(1):7026–7071, 2017.

[64] Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

[65] HP Vinutha, B Poornima, and BM Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and Decision Sciences*, pages 511–518. Springer, 2018.

[66] Luis Von Ahn. Human computation. In *2008 IEEE 24th international conference on data engineering*, pages 1–2. IEEE, 2008.

[67] Panpan Xu, Nan Cao, Huamin Qu, and John Stasko. Interactive visual co-cluster analysis of bipartite graphs. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 32–39. IEEE, 2016.

[68] Rui Xu and Don Wunsch. *Clustering*, volume 10. John Wiley & Sons, 2008.

[69] Ritu Yadav and Anuradha Sharma. Advanced methods to improve performance of k-means algorithm: A review. *Global Journal of Computer Science and Technology*, 12(9): 47–52, 2012.

[70] Odilia Yim and Kylee T Ramdeen. Hierarchical cluster analysis: comparison of three linkage measures and application to psychological data. *The quantitative methods for psychology*, 11(1):8–21, 2015.

[71] Qiaoping Zhang and Isabelle Couloigner. A new and efficient k-medoid algorithm for spatial clustering. In *International conference on computational science and its applications*, pages 181–189. Springer, 2005.