Stockholms universitet

Tech conference attendee segmentation using data-driven methods: Identifying clusters based on patterns of engagement

Georgios Lamprou

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2022:1
Matematisk statistik
Februari 2022

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Master Thesis **2022:1**
http://www.math.su.se

Tech conference attendee segmentation using data-driven methods: Identifying clusters based on patterns of engagement

Georgios Lamprou*

February 2022

Abstract

It is becoming increasingly common for companies to try to understand their customers and take various actions using data-driven methods. The data availability, the increased storage capacities and the improvements in computing have all contributed towards this direction. Web Summit, an Irish technology conference company, is no exception and has taken advantage of these developments.

In this thesis, an optimal division of attendees from three conferences into segments was the desired outcome and cluster analysis was the chosen approach. The segments would consist of homogeneous sub-populations which share common behavioural aspects regarding patterns of engagement per conference. The company's mobile app was the source of the various engagement metrics, which were subsequently aggregated.

To achieve the above, three clustering methods were chosen to be fitted, validated and compared. Those methods were K-Medoids, Agglomerative clustering with average-linkage and ("fuzzy") HDBSCAN. The data were transformed before clustering with UMAP, a fairly new dimensionality reduction method, after a simulation study which showed that UMAP-assisted clustering provides a performance advantage.

It was concluded that overall Agglomerative clustering with average-linkage had a small advantage over the other methods. The cluster centroids revealed a shared pattern that underlines all clusters, namely the increase in engagement as the event approaches and the abrupt decrease of it when it ends. Most importantly it revealed spikes in engagement on different days, which is what makes the clusters distinct.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: education@glamprou.eu. Supervisor: Taras Bodnar.

Acknowledgments

First and foremost, I would like to thank my supervisor from the Department of Mathematics, Prof. Taras Bodnar for his guidance and his patience. The latter was much needed considering the challenging task of remote supervision. I want to thank my external supervisor Dr. Steven Tobin who, by acting on behalf of Web Summit, facilitated this project in many ways. I would like to express my gratitude to Dr. Jan-Olov Persson, responsible for coordinating the course, for timely responding to all my queries. Finally, I am thankful to Ann for all the support throughout this journey.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

AMI Adjusted Mutual Information. 23

ARI Adjusted Rand Index. 22

ARMA Autoregressive Moving Average. 12

AWS Amazon Web Services. 10

c Completeness. 23

C Ground truth classes. Individual classes are represented as $C_i, i = 1, .., r$. 22, 23, 24

CE Cross entropy. 19

CR Curve. 20

d Distance. 12, 14, 15, 16, 20, 21, 25

D Degree matrix. 19, 20

d_c Core Distance in the context of HDBSCAN algorithm. 16, 17

d_{DTW} Dynamic Time Warping Distance. 13

d_E Euclidean Distance. 12

$d_{\min}$ Minimum distance. 20

d_{mr} Mutual Reachability Distance. 16

d_r Correlation Distance. 13

DB Davies-Bouldin index. 25

$DBCV$ Density-Based Clustering Validation. 26, 36

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 16

Δ_k The intra-cluster distance of cluster K_k. 25

DI Dunn index. 25

DSC Density Sparseness of a Cluster. 26

$DSPC$ Density Separation of a Pair of Clusters. 26

E Expected value. 22, 23, 24

E_p Expected value with respect to the distribution p. 19

EC2 Amazon Elastic Compute Cloud. 10

FMI Fowlkes-Mallows index. 23

G_{ji} The total gain or contribution in the context of the BUILD phase of PAM algorithm. Partial gain/contribution is represented as g_{ji}. 14

m_{pts} A smoothing factor for density estimates. 16

$G_{m_{pts}}$ Mutual Reachability Graph. 16, 17

h Homogeneity. 22, 23

H Shannon's entropy. 22, 23

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise. 13, 16

HS Hopkins statistic. 21

K Clustering implementation. Individual clusters are represented as $K_i, i = 1, ..., s$. 15, 17, 18, 22, 23, 24, 25, 26

κ The centroid of the cluster, e.g. κ_1 would be the centroid of the cluster K_1. 25

l Number of time series points. Alternatively the number of features. 11

λ The inverse of the split distance in the HDBSCAN context. 17, 18

LTS Long-term support. 10

MI Mutual Information. 23, 24

MST Minimum Spanning Tree. 17

n Number of common observations between two partitions. 22, 23, 24

N Number of data points. 11, 22, 23, 24

O Noise. 26

OS Operating System. 10

P Probability. 18

PAM Partitioning Around Medoids. 14

ϕ_{ji} The difference between the distance of x_j with its nearest selected observation and the distance of x_j and x_i in the context of the K-medoids algorithm. 14

π Warping path. 13

ψ The merge height of an observation in the HDBSCAN context. 18

r Pearson's correlation coefficient. 13

RI Rand Index. 22

T_{ih} The total contribution in the context of the SWAP phase of PAM algorithm. Partial contribution is represented as t_{jih}. 14

UMAP Uniform Manifold Approximation and Projection. 2, 19, 20

UPGMA Unweighted Pair Group Method with Arithmetic mean. 15

x Time series. Individual time points are represented as $x_{ij}, j = 1, ..., l$, e.g. $x_1 = \{x_{11}, x_{12}, ..., x_{1l}\}$. 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25

X A set of time series. 14, 16, 20, 21

1. Introduction

1.1 Web Summit – a tech conference company

Based on information on its website [31], Web Summit is a tech conference company founded in 2009 in Dublin, Ireland. Since then, it has exhibited significant growth both in number of attendees and partnerships. It has hosted primarily in-person events in Dublin, Lisbon, Toronto, Hong Kong, New Orleans, Bangalore and cities like Tokyo and Kuala Lumpur will be added to the list soon. The company's flag event is named after it (Web Summit) and it is the biggest in size and publicity. Other events are Collision, RISE, SURGE (inactive) and MoneyConf (part of the Web Summit event now).

The company's events are a good place for startups, investors and simple tech enthusiasts to meet and explore interests and opportunities. Market research, lead generation, networking and professional development are some key benefits of attendance. The Web Summit event has been characterized as a leading technology conference worldwide [8].

In 2015, Web Summit launched a mobile app. The latter would be the place where the conference ticket would be stored. Moreover, it had several other features such as online communication among attendees, schedule and personalized recommendations for conference sessions. The app has been essential throughout the years for a complete event experience.

In 2020, the COVID-19 pandemic affected, among others, the way people interact socially and professionally. It also encouraged the acceleration of digitalisation in the Western world. These facts had, of course, implications on how mass public events are held. Following this trend, Web Summit decided to organise exclusively online events for the first time. To achieve that, it enhanced significantly its software infrastructure and created a platform on which these virtual events can and have already taken place.

The aforementioned situation makes even more urgent the need for understanding the company's customer base and, particularly, conference attendees. This would facilitate interventions on every level improving the quality of the product and as a result the attendee satisfaction.

1.2 Customer segmentation

Tsiptsis and Chorianopoulos [51] defined customer segmentation as "the process of dividing customers into distinct, meaningful, and homogeneous subgroups based on various attributes and characteristics" [51, p. 189]. It started being used to address the drawbacks of mass marketing, particularly the inability to follow the developments in the market in the 20th century, such as the increasing fragmentation and cost of advertising [29, p. 240].

The objectives of customer segmentation can be many. Companies can implement it to understand their customers, create new products and services or improve existing ones, offer discounts, adjust their marketing strategy, adjust their resource allocation and others [51, p. 190]. The segments make the customer base more manageable and facilitate decision making [51, p. 190].

Segmentation can take various forms and selecting one of them can be dependent on the objective and/or the data availability. Kotler and Keller [29] identify geography, demographics, psychology and behavior as the main pillars on which this process can be based [29, p. 247], but there can, also, be combinations of them.

1.3 The goal of this project

In this project, the focus will be on behavioral segmentation and by behavior we mean patterns of user engagement before, during and after the three events in question. The goal is to identify segments of conference attendees, who exhibit similar patterns of engagement time-wise. The metrics of engagement will be derived exclusively from the mobile app usage.

This data-driven approach, apart from its descriptive value, will be proven useful for potential time-dependent attempts to engage with groups of attendees either on a marketing or product-related basis (e.g. recommender system). However, segment profiling and deployment issues will not be a part of this project.

1.4 Software and hardware

The source code for this project is being written in Python. The data analysis took place in JupyterLab running in a m4.large EC2 instance on AWS, having Ubuntu 20.04.02 LTS as OS. The Python packages used, aside from the ones in the Python Standard Library, were hdbscan [34], scikit-learn [43], scikit-learn-extra, pandas [50] [39], matplotlib [25], NumPy [22], UMAP [36], validclust, pyclustertend, Yellowbrick [6], kneed [48] and dill [38] [37].

2. Methods

2.1 Cluster analysis

Cluster analysis or clustering is a set of techniques used for identifying groups in data [27, p. 1]. It differs from classification in the sense that the algorithms used are considered to be unsupervised learning. In classification, the class labels are already known, while in clustering they need to be discovered.

Assigning objects to groups is an important aspect of human behavior, starting from the early childhood [27, p. 1]. That is to address the problem of managing the countless objects (real or abstract) humans encounter every day [2, p. 1]. Moreover, it has been used in various fields, such as biology, social science, geoscience, medicine and engineering [2, p. xi]. Reasons to use it can be either descriptive or predictive or both.

Everitt et al. [17, pp. 7, 8] claim that it is difficult to give a clear definition of the cluster, though measures related to cohesion and separation have been used to form one. They acknowledge the fact that clusters are often identified visually and sometimes there is a disagreement between that and methods of cluster analysis. According to them, that is often the case when it comes to uniform datasets, where a method can impose a non-existent structure. However, even in cases like that, it might be meaningful to dissect the data [17, p. 8].

Kaufman and Rousseeuw [27, pp. 3, 4] identify two types of input data that can be used for clustering. The first one, according to them, is a $N \times l$ matrix where N is the number of observations and l is the number of features. Features are carefully selected, usually on empirical grounds. The second is a $N \times N$ matrix of the similarities or dissimilarities of observations [27, pp. 3, 4].

According to Han et al. [21, pp. 448–451] there are four general categories of clustering methods. They identify: 1. Partitioning methods, which separate data into different partitions on a single-level basis, with a predefined number of clusters; 2. Hierarchical methods, which separate data on a multi-level basis and can be performed on a bottom-up or top-down way; 3. Density-based methods, which do not work directly on the distance, but on the "neighborhood" of data points and can exclude many of them by treating them as noise; 4. Grid-based methods which create a grid data structure and perform clustering on that, demonstrating a performance advantage compared to the other categories of methods.

2.2 Time-series clustering

Aghabozorgi et al. [1] definition of time-series clustering does not differ from the classical one. They state, however, that time-series have special characteristics and pose different challenges, with respect to size, dimensionality and similarity. There are, also, unique aspects in objectives (e.g. anomaly detection, dynamic changes, etc.) and taxonomy (e.g. clustering of multiple entire time-series or parts of one time-series) [1].

According to Warren Liao [53] time-series can either be handled as they are by fitting an appropriate similarity/dissimilarity measure or converted to a feature vector, using dimensionality reduction techniques or even model parameters. The nature of the data play a role on how they will be handled, particularly in terms of the quality of the sampling, the number of time-dependent variables, the nature of the numbers and the length (varying or not) [53].

2.3 Similarity/distance

The choice of time-series similarity/distance metric needs always careful consideration, because it is tightly connected to the objectives of clustering. Bagnall and Janacek [3] identify similarity in time, shape and change as the three categories that guide the choice of a similarity/distance metric. Similarity in time can be achieved with distance metrics like the correlation distance or the Euclidean distance on the normalised time-series, while Dynamic Time Warping, introduced by Bellman and Kalaba [5], is a popular method for similarity in shape [3]. Similarity in change is a bit different, since time-series models, such as ARMA, have to be utilized [3].

The distance between two time-series $x_1 = \{x_{11}, x_{12}, ..., x_{1l}\}$ and $x_2 = \{x_{21}, x_{22}, ..., x_{2l}\}$ can be generally defined as follows [1]:

$$d(x_1, x_2) = \sum_{i=1}^{l} d(x_{1i}, x_{2i}) \tag{2.1}$$

The Euclidean distance is [53]:

$$d_E(x_1, x_2) = \sqrt{\sum_{i=1}^{l} (x_{1i} - x_{2i})^2} \tag{2.2}$$

and it is the most widely used. Data normalization/standardization is often required before fitting. It has variations, such as the squared Euclidean distance and

generalizations such as the Minkowski distance. The correlation distance is [53]:

$$d_r(x_1, x_2) = 1 - r_{x_1,x_2} = 1 - \frac{\sum_{i=1}^{l}(x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{\sqrt{\sum_{i=1}^{l}(x_{1i} - \bar{x}_1)^2}\sqrt{\sum_{k=1}^{l}(x_{2i} - \bar{x}_2)^2}} \tag{2.3}$$

where r is the Pearson's correlation coefficient. Since the latter is a value between 0 and 1, the correlation distance can take values between 0 and 2. Given the lattice $\mathcal{L}_{l,l} = \{1, ..., l\} \times \{1, ..., l\}$, the Dynamic Time Warping distance is [26]:

$$d_{DTW}(x_1, x_2) = \min_{\pi \in \Pi} \sqrt{\sum_{(i,j)\in\pi} (x_{1i} - x_{2j})^2} \tag{2.4}$$

where $\pi = \{\pi_1, \pi_2, ..., \pi_z\}$ is a warping path and Π is the set of all warping paths in $\mathcal{L}_{l,l}$. A warping path is a sequence of points $\pi_k = (i_k, j_k)$ in $\mathcal{L}_{l,l}$ which satisfy the follwing conditions [26]

- (boundary condition) $\pi_1 = (1, 1)$ and $\pi_z = (l, l)$
- (step condition) $\pi_{k+1} - \pi_k \in \{(1, 0), (0, 1), (1, 1)\}, \forall k \in [z - 1]$

From the above, only the Euclidean distance is a metric, since the rest do not satisfy the triangular inequality. This does not, however, prevent them from being used extensively. Moreover, there is no restriction on the input data for these distances. Finally, the choice of similarity/distance metric should be unique when comparing different clustering methods using exclusively internal validity indices, since otherwise it is like comparing methods on different data [1].

2.4 Clustering methods

The list of methods that have been used for time-series clustering is rather large. They range from original ones, such as Fuzzy c-means, Agglomerative clustering, k-means, K-Medoids and SOM to adjusted, such as Modified relocation clustering, modified k-means, Modified CAST [1] [53]. In the following subsections, we will expand on three algorithms, specifically K-Medoids, Agglomerative Clustering and HDBSCAN, which were chosen for three reasons: 1. Each one of them represents a broader class of models, 2. They all can be used with distance measures and 3. They produce results relatively fast.

2.4.1 K-Medoids

K-Medoids is a partitioning-based method. Medoids are actual observations in contrast with K-Means where the representatives are artificial. The name of the

method was given by Kaufmann and Rousseeuw [28] who, also, introduced one of its implementations, the Partitioning Around Medoids (PAM) algorithm. The method aims to allocate observations to clusters based on the minimum distance from their medoids [28]. In K-Medoids, as in K-Means, the number of clusters has to be provided.

Kaufman and Rousseeuw [27, pp. 102–104] describe PAM, which consists of the phases BUILD and SWAP that can be summarized in algorithm 1 and algorithm 2. Let X_s be the selected and X_u the unselected observations and let x_m the observation whose sum of distances to all other observations is the smallest [27, p. 102].

Algorithm 1: Find initial k medoids (BUILD)

1. Add observation x_m to X_s.
2. Pick observation $x_i \in X_u$.
3. Pick $x_j \in X_u - \{x_i\}$ and calculate $\phi_{ji} = d(x_j, x_{1nn_j}) - d(x_j, x_i)$, where d the distance and x_{1nn_j} the nearest selected observation to x_j.
4. Calculate the partial contribution $g_{ji} = max(\phi_{ji}, 0)$. If $g_{ji} = \phi_{ji}$ then x_j will play a role in the selection process of x_i.
5. Calculate the total contribution $G_{ji} = \sum_j g_{ji}$.
6. Select the x_i that maximizes G_{ji}.
7. Repeat steps 2-6 until k medoids have been found.

Now let x_{2nn_j} be the second nearest selected observation to x_j and (x_i, x_h) be all pairs of observations where x_i has been selected, while x_h has not [27, p. 103].

Algorithm 2: Update k medoids (SWAP)

1. Pick observation $x_j \in X_u - \{x_h\}$.
2. Calculate

$$t_{jih} = \begin{cases} min(d(x_j, x_h) - d(x_j, x_{1nn_j}), 0), & d(x_j, x_i) > d(x_j, x_{1nn_j}) \\ min(d(x_j, x_h), d(x_j, x_{2nn_j})) - d(x_j, x_{1nn_j}), & d(x_j, x_i) = d(x_j, x_{1nn_j}) \end{cases}$$

   where t_{jih} the partial contribution to the swap.
3. Calculate the total contribution to the swap $T_{ih} = \sum_j t_{jih}$.
4. Select x_i and x_h that minimize T_{ih}.
5. If $\min T_{ih} < 0$, the swap takes place and algorithm restarts. If $\min T_{ih} \geq 0$ the swap is meaningless and the algorithm stops.

In some cases, such as when we try to determine the number of clusters, speed is crucial. An alternate method which will replace algorithm 2 can be used [42].

Algorithm 3: Update k medoids and assign observations (Alternate)

1 Assign observations to their nearest medoid.
2 Calculate the sum of distances between all observations and their medoids.
3 For each cluster, find the observation that minimizes the sum of distances between itself and all other observations in the cluster. Make this observation the new medoid of the cluster.
4 Repeat steps 1 and 2. If the number we get from step 2 is equal to the previous calculation, the algorithm stops. If not, go back to step 3.

2.4.2 Agglomerative clustering: Average-linkage

Agglomerative clustering is one of the two main methods of hierarchical clustering, the other one being the divisive clustering [32]. While the divisive clustering utilizes a "top-down" approach, the agglomerative clustering utilizes a "bottom-up" approach. This, basically, means that we start with as many clusters as the number of observations and we gradually merge those to larger clusters, until we reach the point of having just one big cluster [32], as shown in algorithm 4.

A choice has to be made about the linkage criterion, since different criteria yield different results. They control how the precomputed pairwise distances determine the distance between groups of observations. Some of those criteria are the Ward's criterion, the complete-linkage, the average-linkage and the single-linkage. Average-linkage or Unweighted Pair Group Method with Arithmetic mean (UPGMA) creates clusters in a way that each observation has smaller average distance to all observations inside its own cluster than the rest of the observations and can be summarized as [49]:

$$d(K_1, K_2) = \frac{1}{|K_1||K_2|} \sum_{x_1 \in K_1} \sum_{x_2 \in K_2} d(x_1, x_2) \tag{2.5}$$

which is the initial distance between clusters K_1, K_2 and

$$d(K_1 \cup K_2, K_3) = \frac{|K_1| d(K_1, K_3) + |K_2| d(K_2, K_3)}{|K_1| + |K_2|} \tag{2.6}$$

which is the distance between clusters after a merge has occurred.

Algorithm 4: Agglomerative clustering

1 Add each observation to a separate cluster and calculate their distance.
2 Merge similar clusters based on the smallest distance d between them.
3 Calculate the distances of the new clusters.
4 Repeat steps 2-3 until a single cluster is formed.

The output of agglomerative clustering is a dendrogram with leaves being the individual observations and nodes being the clusters in all levels, with root node the single upper level cluster. The naive implementation of the algorithm has time complexity $O(n^3)$, but there exist different approaches which can reduce that significantly [20].

2.4.3 HDBSCAN

HDBSCAN or Hierarchical Density-Based Spatial Clustering of Applications with Noise is an extension of DBSCAN. Introduced by Campello et al. [10], the algorithm aims to address issues of other density-based methods, including but not limited to the need to have multiple density thresholds for clusters of different densities and structure, to identify the most significant clusters and to minimize the need for sensitive input parameters.

Summarizing how HDBSCAN works (algorithm 5), requires the definition of certain notions [10]:

1. **Core Distance (d_c)**: The distance between an observation x_i in X and its m_{pts}-nearest neighbor, where m_{pts} a smoothing factor for density estimates.

2. **ϵ-Core Object**: An observation x_i in X, whose core distance is smaller or equal to ϵ.

3. **Mutual Reachability Distance (d_{mr})**: The distance

$$d_{mr} = \max\{d_c(x_i), d_c(x_j), d(x_i, x_j)\} \tag{2.7}$$

   where x_i, x_j in X.

4. **Mutual Reachability Graph ($G_{m_{pts}}$)**: A graph consisted of the observations of X as vertices, with edge weights being the d_{mr} of all pairs of observations.

5. **Minimum Spanning Tree**: A subset of the edges of a connected undirected graph that connects all vertices together with minimal weight [41].

Algorithm 5: Construct the HDBSCAN hierarchy

1. Calculate d_c for all observations.
2. Calculate a Minimum Spanning Tree (MST) of $G_{m_{pts}}$.
3. Add a "self edge" to each vertex with weight being the d_c of the relevant observation and, thus, extend MST.
4. Add all observations to the same cluster. This would be the root of the dendrogram.
5. Set the dendrogram scale value equal to the largest weight of the existing edges.
6. Remove the edge with the largest weight from the extended MST.
7. Assign cluster labels to the connected components linked to the removed edge if they have one or more edges. If that is not the case, classify the components as noise. Alternatively, if a connected component has fewer observations than a given minimum cluster size, then it is classified as noise and the cluster split is not taking place. If, on the other hand, the split results to two clusters with number of observations larger or equal to the minimum cluster size, then the split is valid.
8. Repeat steps 5-7 until all edges are removed.

We can extract flat clusters from the hierarchical tree. This can be achieved by either using an excess of mass method or just choosing the clusters at the leaves of the tree. The former aims to find the most persistent clusters and uses the concept of stability [34] [10]

$$S(K_i) = \sum_{x_j \in K_i} (\min(\lambda_i^{death}, \lambda_{ij}) - \lambda_i^{birth}) \tag{2.8}$$

where K_i a cluster, λ is the inverse of the split distance, λ_i^{birth} the value of λ at which the cluster was created, λ_i^{death} the value of λ at which the cluster was subsequently divided to more clusters and λ_{ij} is the value when the point x_j stopped belonging to that cluster. Initially, we consider all leaves as the selected flat clusters and we move up towards the root of the tree. In case the sum of stabilities of the child clusters is larger than the cluster's stability, we set the latter to be equal to the former. If the opposite is the case, then we select that cluster and disregard its child clusters.

It is possible to implement a "fuzzy" version of HDBSCAN where instead of observations being assigned clustered memberships they are assigned probabilities of membership to each cluster [34]. This is useful, especially, when we want to assign all observations classified as noise to their nearest cluster. This method is described in algorithm 6, algorithm 7 and algorithm 8 [34].

Algorithm 6: Distance-based membership

1 Select a cluster K_i and get its leaf clusters.
2 For each leaf, find the observations with the largest λ. These observations are the exemplars of the cluster.
3 Repeat steps 1 and 2 until determining exemplars for all clusters.
4 Calculate the inverse minimum distance of every observation to the cluster exemplars and divide by the sum of these distances for all clusters. The results are the cluster membership scores of the observations.

Algorithm 7: Outlier-based membership

1 Select a cluster K_i and get its leaf clusters. Find its $\lambda_i^{\max}$, namely the largest λ of its leaves.
2 For each observation x_j, find its merge height ψ_{ij} with K_i.
3 Repeat steps 1 to 2 until determining all $\lambda_i^{\max}$ and ψ_{ij}.
4 For each observation, calculate

$$\rho_{ij} = \frac{\lambda_i^{\max}}{\lambda_i^{\max} - \psi_{ij}} \tag{2.9}$$

5 For each observation, calculate the exponential of all ρ_{ij} and divide by the sum of ρ_{ij} for all clusters. The latter are the cluster membership scores of the observations.

Algorithm 8: Combined membership

1 For each observation, multiply the cluster membership vectors found by algorithm 6 and algorithm 7 and divide by the sum of the resulting vector. That is the conditional probabilities of observations are in each cluster based on them actually belonging in a cluster. They can be defined as:

$$P(x \in K_i | \exists j : x \in K_j) = \frac{P(x \in K_i, \exists j : x \in K_j)}{P(\exists j : x \in K_j)} \tag{2.10}$$

We have $P(x \in K_i) = P(x \in K_i, \exists j : x \in K_j)$, since $P(\exists j : x \in K_j) = 1$ if $x \in K_i$
2 Estimate $P(\exists j : x \in K_j)$ by calculating $\frac{\psi_{ij}}{\lambda_i^{\max}}$ for each observation.
3 Compute $P(x \in K_i)$.

2.5 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a relatively new technique used for dimensionality reduction and subsequent visualization or sometimes as input to clustering or other algorithms [35]. The theoretical underpinnings of UMAP, mainly topology and category theory, are outside the scope of this project. The readers who are interested in them are referred to [30], [33] and [45]. McInnes et al. [35] describe the algorithm as shown in algorithm 9. Below we briefly describe some necessary terms [35]:

1. **n-simplex**: A convex hull of $n+1$ independent points in the Euclidean space [12]. For example, the 0-simplex is a point, the 1-simplex is a line segment, the 2-simplex is a triangle and so forth [12].

2. **Simplicial Set**: Originating in Eilenberg and Zilber [16], it is a generalization or abstraction of simplicial complex, the latter being a set of simplices such that the faces of the simplices belong to it and the intersection of any two simplices is a face of both of them. A simplicial set is a mapping of the simplex category to the set category.

3. **n-skeleton**: A simplicial subcomplex with dimension n.

4. **k-nearest neighbors**: A classification and regression method originating in Fix and Hodges [18], with the central idea that neighboring observations "vote" for the values of observations in question.

5. **Weighted Adjacency Matrix**: A square matrix A whose elements are the weights of the edges of a graph $\bar{\Gamma}$, when the edges exist and zero otherwise.

6. **Degree Matrix**: A square matrix D of a graph $\bar{\Gamma}$ with elements [11]

$$D_{ij} = \begin{cases} deg(x_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

   where $deg(x_i)$ the number of either the incoming or outgoing edges at each vertex if the graph is directed or the number of edges at each vertex if the graph is undirected (loops count as 2).

7. **Cross entropy**: It measures the amount of information needed for a probability distribution p to identify samples from a probability distribution q over the same set and it can be defined as:

$$CE(p,q) = -E_p\left[\log q(x)\right] \tag{2.12}$$

Algorithm 9: UMAP

1 For x_i in X and $X_{i_{nn}} = \{x_{i_j} | j = 1, ..., k\}$ the k nearest neighbors of x_i, construct a local fuzzy simplicial set by creating a weighted directed graph $\bar{\Gamma} = (X, \Omega, w)$ where Ω is the set of the directed edges (x_i, x_{i_j}) with weights:

$$w_{ij} = \exp[\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}] \tag{2.13}$$

where $\rho_i = \min(d(x_i, x_{i_j}) | 1 \leq j \leq k, d(x_i, x_{i_j} > 0)$ and σ_i is a normalization factor which should satisfy the equality $\sum_i^k w_{ij} = \log_2 k$, with k being the neighborhood size. Convert $\bar{\Gamma}$ into an undirected weighted graph by calculating a new adjacency matrix

$$A_s = A + A^T - A \circ A^T \tag{2.14}$$

where A is the weighted adjacency matrix of $\bar{\Gamma}$ and $\circ$ is the Hadamard product. The matrix A_s is the 1-skeleton of the topological representation.

2 Initialize graph layout using spectral embedding:

$$Y = \Lambda[1..\tau + 1] \tag{2.15}$$

where τ the desired reduced dimension, Λ are the eigenvectors of $D^{1/2}(D - A_s)D^{1/2}$ and D the degree matrix of the undirected graph.

3 Optimize embedding by minimizing the quantity:

$$-\sum_{ij} w_{ij}^* \log(v_{ij}) + (1 - w_{ij}^*) \log(1 - v_{ij}) \tag{2.16}$$

which is derived by the fuzzy set cross entropy. The weight w_{ij}^* corresponds to the undirected graph from the previous steps and v_{ij} is the corresponding weight in the low-dimensional space and it is approximated by

$$v_{ij} = \frac{1}{1 + ad(y_i, y_j)^{2b}} \tag{2.17}$$

where a, b are calculated by non-linear least squares fitted to the curve

$$CR_{ij} = \begin{cases} 1, & \text{if } d(y_i, y_j) \leq d_{\min} \\ \exp(-d(y_i, y_j) - d_{\min}), & \text{otherwise} \end{cases} \tag{2.18}$$

where $d_{\min}$ the provided minimum distance between embedded points.

2.6 Validation

Clustering validation is the process of assessing how the clustering methods perform on real data. It can be used either for determining the number of clusters and other parameters for a single method or for comparing the performance of different methods. It can, also, be used for assessing the clustering tendency in the data.

Validation indices can be classified as either external or internal. External indices can only be used if a "ground truth" of the clusters is known. This is not the case, however, for the majority of the problems, since clustering is mainly used as an exploratory technique for uncovering an unknown structure in the data. The Rand index, the Jaccard index, the Fowlkes-Mallows index are some external indices. On the other hand, we have the internal indices which do not require any prior knowledge about the data structure. They assess how well the produced clusters match that structure. They include but are not limited to the Silhouette score, the Dunn index, the Davies-Bouldin index.

Internal indices measure 1) compactness, which is how close to each other the cluster elements are and 2) separability, which is how far away from each other different clusters are [44]. There should be careful interpretation of the indices' values, since some of them are positively biased towards certain clustering techniques.

2.6.1 Cluster tendency - Hopkins statistic

The Hopkins statistic originates in Hopkins and Skellam [23] as part of an alternative method to quadrats for the determination of distribution of plants. Banerjee and Dave [4] used it as a method to assess clustering tendency and defined it as

$$HS = \frac{\sum_{i=1}^{M} d(y_i, x_{y_{i_1}})}{\sum_{i=1}^{M} d(y_i, x_{y_{i_1}}) + \sum_{i=1}^{M} d(x_i^*, x_{x_{i_1}^*})} \tag{2.19}$$

where $x_{y_{i_1}}$ is the nearest neighbor of y_i in X and $x_{x_{i_1}^*}$ is the nearest neighbor of x_i^* in X, while $X = \{x_1, x_2, ..., x_N\}$ the points in l-dimensional space, $X^* = \{x_1^*, x_2^*, ..., x_M^*\}$ a random sample of X , with $M << N$, and $Y = \{y_1, y_2, ..., y_M\}$ points in l-dimensional space sampled from a uniform random distribution. In many cases, the value $1 - HS$ is taken into account instead. If the latter is close to 0 then the data exhibit cluster tendency, while if it is 0.3 and above, then the data are near uniform.

2.6.2 External validity indices

Let N be the number of data points, $K = \{K_1, K_2, ..., K_s\}$ be a clustering implementation, $C = \{C_1, C_2, ..., C_r\}$ be the ground truth classes, $n_{ck} = \{ck : c = 1, .., r | k = 1, ..., s\}$ be the number of common observations between the elements of C and K, as shown in the contingency table

C/K	K_1	K_2	...	K_s	sum
C_1	n_{11}	n_{12}	...	n_{1s}	n_{1*}
C_2	n_{21}	n_{22}	...	n_{2s}	n_{2*}
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
C_r	n_{r1}	n_{r2}	...	n_{rs}	n_{r*}
sum	n_{*1}	n_{*2}	...	n_{*s}	

Adjusted Rand Index

The Adjusted Rand Index is defined as [24]:

$$ARI = \frac{\sum_{ck} \binom{n_{ck}}{2} - [\sum_c \binom{n_{c*}}{2} \sum_k \binom{n_{*k}}{2}] \binom{n}{2}}{\frac{1}{2}[\sum_c \binom{n_{c*}}{2} + \sum_k \binom{n_{*k}}{2}] - [\sum_c \binom{n_{c*}}{2} \sum_k \binom{n_{*k}}{2}]/\binom{n}{2}} \tag{2.20}$$

Another formulation of the index is [24]

$$ARI = \frac{RI - E(RI)}{\max RI - E(RI)} \tag{2.21}$$

where $RI = \frac{a+b}{\binom{n_{ck}}{2}}$ is the Rand index, a being the number of pairs of observations belonging in the same subsets in both C and K, while b is the number of pairs of observations belonging in different subsets in both C and K.

The ARI takes values from -1 to 1. The higher the values the more the two clustering implementations look alike, while a value close to 0 signifies randomness in the assignment of observations to clusters. Negative values signify less than expected similarity between the two clusterings.

Homogeneity - Completeness - V-measure

These indices relate to Shannon's entropy. Homogeneity is defined as [46]

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{2.22}$$

where

$$H(C|K) = -\sum_{k=1}^{s}\sum_{c=1}^{r}\frac{n_{ck}}{N}\log\frac{n_{ck}}{\sum_{c=1}^{r} n_{ck}} \tag{2.23}$$

and

$$H(C) = -\sum_{c=1}^{r}\frac{\sum_{k=1}^{s} n_{ck}}{r}\log\frac{\sum_{k=1}^{s} n_{ck}}{r} \tag{2.24}$$

Completeness is symmetric to homogeneity and defined as [46]

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{2.25}$$

where

$$H(K|C) = -\sum_{c=1}^{r}\sum_{k=1}^{s}\frac{n_{ck}}{N}\log\frac{n_{ck}}{\sum_{k=1}^{s} n_{ck}} \tag{2.26}$$

and

$$H(K) = -\sum_{k=1}^{s}\frac{\sum_{c=1}^{r} n_{ck}}{r}\log\frac{\sum_{c=1}^{r} n_{ck}}{r} \tag{2.27}$$

Finally, V-measure is [46]

$$V_\beta = \frac{1+\beta hc}{\beta h + c} \tag{2.28}$$

where β is a weight to determine the importance of homogeneity or completeness over the other.

Fowlkes-Mallows index

The Fowlkes-Mallows index is [19]

$$FMI = \frac{\sum_{c=1}^{r}\sum_{k=1}^{s} n_{ck}^2 - N}{(\sum_{k=1}^{s} n_{*k}^2 - N)(\sum_{c=1}^{r} n_{c*}^2 - N)} \tag{2.29}$$

It can range from 0 to 1, with the latter signifying a good matching between the partitions.

Adjusted Mutual Information

The Adjusted Mutual Information index is [52]

$$AMI = \frac{MI(C,K) - E\{MI(C,K)\}}{avg\{H(C), H(K)\} - E\{MI(C,K)\}} \tag{2.30}$$

where

$$MI(C,K) = \sum_{c=1}^{r}\sum_{k=1}^{s} \frac{n_{ck}}{N} \log \frac{n_{ck}/N}{n_{c*}n_{*k}} \tag{2.31}$$

$$E\{MI(C,K)\} = \sum_{c=1}^{r}\sum_{k=1}^{s} \sum_{n_{ck}=(n_{c*}+n_{*k}-N)^{+}}^{\min(n_{c*},n_{*k})} \frac{n_{ck}}{N} \log \frac{Nn_{ck}}{n_{c*}n_{*k}} \times \frac{n_{c*}!n_{*k}!(N-n_{c*})!(N-n_{*k})!}{N!n_{ck}!(n_{c*}-n_{ck})!(n_{*k}-n_{ck})!(N-n_{c*}-n_{*k}+n_{ck})!} \tag{2.32}$$

where MI the Mutual Information index. The index ranges from -1 to 1, where the latter signifies perfect overlapping and we get a value of 0 when the Mutual Information index is the same with its expected value.

2.6.3 Internal validity indices

Let again $K = \{K_i : i = 1, 2, ..., s\}$ be a clustering implementation.

Distortion

Technically not a validity index, distortion is the Within-Cluster Sum of Squares

$$WCSS = \sum_{i=1}^{s}\sum_{v=1}^{N_i} d(x_v, \bar{x}_i)^2 \tag{2.33}$$

where $x_v = \{x_{vw}|w = 1, ..., l\}$ a data point, N_i the number of observations in cluster K_i and $\bar{x}_i$ the mean value of data points in the cluster K_i, the distance d is usually Euclidean. The distortion scores are typically used in the Elbow method.

Calinski-Harabasz index

The Calinski-Harabasz index is [9]

$$CH_k = \frac{tr(B_k)}{tr(W_k)} \frac{N-s}{s-1} \tag{2.34}$$

where

$$B_k = \sum_{i=1}^{s} |K_i|(\kappa_i - \kappa_G)(\kappa_i - \kappa_G)^T \tag{2.35}$$

$$W_k = \sum_{i=1}^{s} \sum_{x \in K_i} (x - \kappa_i)(x - \kappa_i)^T \tag{2.36}$$

where κ_G is the global centroid.

Davies-Bouldin index

The Davies-Bouldin index is [13]

$$DB = \frac{1}{s}\sum_{1}^{s}\max_{i\neq j} R_{ij} \tag{2.37}$$

where

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \tag{2.38}$$

$$S_i = \left(\frac{1}{|K_i|}\sum_{j=1}^{|K_i|}|x_j - \kappa_i|^q\right)^{1/q} \tag{2.39}$$

$$M_{ij} = \left(\sum_{k=1}^{m}|\kappa_{ik} - \kappa_{jk}|^p\right)^{1/p} \tag{2.40}$$

where $x_j = \{x_{j1}, x_{j2}, ..., x_{jm}\}$ an observation belonging to cluster K_i and $\kappa_i = \{\kappa_{i1}, \kappa_{i2}, ..., \kappa_{im}\}$ is the centroid of the cluster K_i. The index takes values larger or equal to zero and the smaller the values the better the score.

Silhouette coefficient

The Silhouette coefficient is defined as [47]:

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, & \text{if}|K_i| > 1 \\ 0, & \text{if}|K_i| = 1 \end{cases} \tag{2.41}$$

where $a(i)$ is the average distance between x_i and the rest of observations in cluster K_A and $b(i) = \min_{K\neq A} d(i, K)$ is the minimum mean distance between i and the rest of the observations in the clusters to which x_i does not belong. The coefficient can take values between -1 and 1, with 1 indicating a good clustering allocation, 0 indicating uncertainty over the clustering allocation and -1 indicating misclassification [47].

Dunn index

The Dunn index is defined as follows [14][15]:

$$DI_s = \frac{\min_{1\leq i<j\leq s} d(K_i, K_j)}{\max_{1\leq k\leq s}\Delta_k} \tag{2.42}$$

where the nominator is minimum inter-cluster distance of K_i and K_j and the denominator the maximum intra-cluster distance of cluster K_k, having the properties:

$$K_i \neq \emptyset$$
$$K_i \cap K_j = \emptyset,\ i \neq j$$
$$\cup_{i=1}^{k} K_i = K$$

The values of the index are between 0 and infinity and larger values indicate better clustering.

DBCV

Density-Based Clustering Validation is an index created by Moulavi et al. [40] and it is defined as:

$$DBCV(K) = \sum_{i=1}^{s} \frac{|K_i|}{|O|} V_K(K_i) \tag{2.43}$$

where O the noise and

$$V_K(K_i) = \frac{\min\limits_{1\leq\ j\leq s, i\neq j}(DSPC(K_i, K_j)) - DSC(K_i)}{\max(\min\limits_{1\leq j\leq s, i\neq j}(DSPC(K_i, K_j)), DSC(K_i))} \tag{2.44}$$

$DSPC$ or Density Separation of a Pair of Clusters being the Minimum Reachability Distance between the nodes of the Minimum Spanning Tree of the corresponding clusters as defined in the subsection 3.4.3 and DSC or Density Sparseness of a Cluster is the minimum weight of the edges of the MST.

3. Simulation study for UMAP-assisted clustering

A simulation study is useful in order to assess if UMAP-assisted clustering enhances the clustering results. We will generate ten artificial datasets of varying size, where the clusters are isotropic Gaussian blobs. We will then fit the clustering methods on the original data and the UMAP embeddings. Finally, we will compare them based on the previously defined external validity indices.

The parameters of the generation process are:

- The number of features, which in our case is chosen to be 21, being the same as the number of points in the actual time series.
- The standard deviation of the clusters which will be a number between 0 and 5.
- The range of values of observations, which will be numbers within -10 to 10.
- The number of clusters which will be a number from 2 to 12. Number 1 is not considered due to the inability of the used implementation of K-Medoids to handle it. Number 12 is set as the upper limit due to performance considerations, when using heuristics to determine the number of clusters.

We will consider two cases. The first is when the number of clusters used in the clustering methods reflects the ground truth classes in the datasets. The second is when we determine the number of clusters heuristically.

Table 3.1: External validity scores for artificial datasets 1-10 with known clusters

	UMAP+K-med.	K-med.	UMAP+Agg. Cl.	Agg. Cl.	UMAP+HDB.	HDB.
Dataset 1						
Adj.Rand	**0.911**	0.494	0.909	0.001	**0.911**	0.604
Completeness	**0.818**	0.414	0.814	0.180	**0.818**	0.508
Fowlkes–Mallows	**0.950**	0.693	0.949	0.661	**0.950**	0.766

Homogeneity	**0.824**	0.502	**0.824**	0.001	**0.824**	0.580
Adj.Mutual Info	0.820	0.454	0.819	0.001	**0.821**	0.542
V-measure	**0.821**	0.454	0.819	0.002	**0.821**	0.542
Dataset 2						
Adj.Rand	**0.559**	0.295	0.534	0.000	-0.000	0.055
Completeness	**0.464**	0.266	0.436	0.091	0.000	0.107
Fowlkes–Mallows	**0.781**	0.654	0.774	0.712	0.708	0.579
Homogeneity	**0.469**	0.265	0.426	0.000	0.000	0.091
Adj.Mutual Info	**0.467**	0.266	0.431	0.000	-0.000	0.098
V-measure	**0.467**	0.266	0.431	0.001	0.000	0.098
Dataset 3						
Adj.Rand	**0.998**	0.993	**0.998**	0.994	**0.998**	0.997
Completeness	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Fowlkes–Mallows	**0.999**	0.997	**0.999**	0.997	**0.999**	0.998
Homogeneity	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Adj.Mutual Info	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
V-measure	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Dataset 4						
Adj.Rand	**0.999**	0.868	**0.999**	0.998	**0.999**	0.997
Completeness	**0.998**	0.915	**0.998**	0.996	**0.998**	0.994
Fowlkes–Mallows	**0.999**	0.889	**0.999**	0.998	**0.999**	0.997
Homogeneity	**0.999**	0.967	**0.999**	0.997	**0.999**	0.994
Adj.Mutual Info	**0.998**	0.940	**0.998**	0.996	**0.998**	0.994
V-measure	**0.998**	0.941	**0.998**	0.996	**0.998**	0.994
Dataset 5						
Adj.Rand	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Completeness	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Fowlkes–Mallows	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Homogeneity	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Adj.Mutual Info	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
V-measure	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Dataset 6						
Adj.Rand	**0.996**	0.969	**0.996**	0.989	**0.996**	0.981
Completeness	**0.992**	0.956	**0.992**	0.981	**0.992**	0.972
Fowlkes–Mallows	**0.997**	0.976	**0.997**	0.992	**0.997**	0.986
Homogeneity	**0.991**	0.956	**0.991**	0.979	**0.991**	0.972
Adj.Mutual Info	**0.991**	0.956	**0.991**	0.980	**0.991**	0.972
V-measure	**0.991**	0.956	**0.991**	0.980	**0.991**	0.972
Dataset 7						
Adj.Rand	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Completeness	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Fowlkes–Mallows	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Homogeneity	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Adj.Mutual Info	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
V-measure	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Dataset 8						
Adj.Rand	**0.998**	0.985	**0.998**	0.947	**0.998**	0.988
Completeness	**0.995**	0.974	**0.995**	0.980	**0.995**	0.978
Fowlkes–Mallows	**0.998**	0.988	**0.998**	0.957	**0.998**	0.990
Homogeneity	**0.995**	0.976	**0.995**	0.923	**0.995**	0.981
Adj.Mutual Info	**0.995**	0.975	**0.995**	0.950	**0.995**	0.979
V-measure	**0.995**	0.975	**0.995**	0.950	**0.995**	0.980
Dataset 9						
Adj.Rand	0.492	0.216	**0.702**	0.000	0.513	0.276
Completeness	0.528	0.336	**0.620**	0.158	**0.620**	0.514
Fowlkes–Mallows	0.594	0.353	**0.766**	0.463	0.647	0.540
Homogeneity	**0.620**	0.401	0.601	0.003	0.489	0.196
Adj.Mutual Info	0.568	0.363	**0.609**	0.001	0.544	0.283
V-measure	0.570	0.366	**0.611**	0.007	0.547	0.284
Dataset 10						
Adj.Rand	**0.984**	0.576	**0.984**	0.917	**0.984**	0.606
Completeness	**0.961**	0.564	**0.961**	**0.961**	**0.961**	0.575
Fowlkes–Mallows	**0.992**	0.762	**0.992**	0.961	**0.992**	0.781
Homogeneity	**0.957**	0.800	**0.957**	0.815	**0.957**	0.798
Adj.Mutual Info	**0.959**	0.662	**0.959**	0.882	**0.959**	0.668
V-measure	**0.959**	0.662	**0.959**	0.882	**0.959**	0.668

From Table 3.1 we can see clearly that in all ten datasets UMAP-assisted clustering performs better, with the exception of the dataset 2 where it performs better only in the Fowlkes–Mallows index.

When it comes to heuristically found number of clusters, we will examine the following datasets:

- Dataset 1, because of the clear superioriority of UMAP-assisted clustering.
- Dataset 2, because UMAP-assisted HDBSCAN performs worse than just simple HDBSCAN.
- Dataset 3, because of the slight advantage of UMAP-assisted clustering .
- Dataset 5, because of the great performance of all algorithms.

Table 3.2: External validity scores for artificial datasets 1, 2, 3, 5. The number of clusters in K-Medoids and Agglomerative clustering is determined by the optimal distortion score (Elbow method), while HDBSCAN has the default parameters.

	UMAP+K-med.	K-med.	UMAP+Agg. Cl.	Agg. Cl.	UMAP+HDB.	HDB.
Dataset 1						
Adj.Rand	0.812	0.736	0.797	0.758	**0.911**	0.758
Completeness	0.856	0.741	**0.861**	0.769	0.818	0.756
Fowlkes–Mallows	0.902	0.862	0.895	0.873	**0.950**	0.873
Homogeneity	0.655	0.565	0.656	0.602	**0.824**	0.579
Adj.Mutual Info	0.742	0.641	0.744	0.675	**0.821**	0.656
V-measure	0.742	0.641	0.744	0.676	**0.821**	0.656
Dataset 2						
Adj.Rand	**0.559**	0.256	0.435	0.000	0.016	0.204
Completeness	0.464	0.210	0.317	**1.000**	0.089	0.153
Fowlkes–Mallows	**0.781**	0.583	0.682	0.713	0.131	0.541
Homogeneity	0.469	0.319	0.493	0.000	**0.579**	0.241
Adj.Mutual Info	**0.467**	0.253	0.385	0.000	0.150	0.187
V-measure	**0.467**	0.253	0.386	0.000	0.155	0.187
Dataset 3						
Adj.Rand	**0.998**	0.993	**0.998**	0.994	**0.998**	0.997
Completeness	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Fowlkes–Mallows	**0.999**	0.997	**0.999**	0.997	**0.999**	0.998
Homogeneity	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Adj.Mutual Info	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
V-measure	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Dataset 5						
Adj.Rand	0.907	0.833	0.907	0.900	**1.0**	**1.0**
Completeness	**1.000**	0.986	**1.000**	**1.000**	**1.0**	**1.0**
Fowlkes–Mallows	0.926	0.872	0.926	0.921	**1.0**	**1.0**
Homogeneity	0.857	0.773	0.857	0.890	**1.0**	**1.0**
Adj.Mutual Info	0.923	0.866	0.923	0.942	**1.0**	**1.0**
V-measure	0.923	0.866	0.923	0.942	**1.0**	**1.0**

We observe in Table 3.2 that overall UMAP-assisted clustering performs better when the number of clusters is determined by the optimal distortion score.

This is particularly the case for Dataset 1. For Dataset 2, we still observe an advantage for all but one indices for simple HDBSCAN, but with improved scores for UMAP-assisted HDBSCAN. The advantage for UMAP-assisted clustering remains in Dataset 3 and the balance is maintained in Dataset 5.

Table 3.3: External validity scores for artificial datasets 1, 2, 3, 5. The number of clusters in K-Medoids and Agglomerative clustering is determined by the optimal Calinski-Harabasz score, while HDBSCAN has the defaults parameters.

	UMAP+K-med.	K-med.	UMAP+Agg.Cl.	Agg.Cl.	UMAP+HDB.	HDB.
Dataset 1						
Adj.Rand	0.812	0.736	0.797	0.758	**0.911**	0.758
Completeness	0.856	0.741	**0.861**	0.769	0.818	0.756
Fowlkes–Mallows	0.902	0.862	0.895	0.873	**0.950**	0.873
Homogeneity	0.655	0.565	0.656	0.602	**0.824**	0.579
Adj.Mutual Info	0.742	0.641	0.744	0.675	**0.821**	0.656
V-measure	0.742	0.641	0.744	0.676	**0.821**	0.656
Dataset 2						
Adj.Rand	**0.559**	0.295	0.534	0.002	0.016	0.204
Completeness	**0.464**	0.266	0.436	0.072	0.089	0.153
Fowlkes–Mallows	**0.781**	0.654	0.774	0.709	0.131	0.541
Homogeneity	0.469	0.265	0.426	0.005	**0.579**	0.241
Adj.Mutual Info	**0.467**	0.266	0.431	0.006	0.150	0.187
V-measure	**0.467**	0.266	0.431	0.010	0.155	0.187
Dataset 3						
Adj.Rand	**0.998**	0.993	**0.998**	0.994	**0.998**	0.997
Completeness	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Fowlkes–Mallows	**0.999**	0.997	**0.999**	0.997	**0.999**	0.998
Homogeneity	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Adj.Mutual Info	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
V-measure	**0.994**	0.984	**0.994**	0.985	**0.994**	0.990
Dataset 5						
Adj.Rand	**1.0**	0.886	**1.0**	**1.0**	**1.0**	**1.0**
Completeness	**1.0**	0.923	**1.0**	**1.0**	**1.0**	**1.0**
Fowlkes–Mallows	**1.0**	0.907	**1.0**	**1.0**	**1.0**	**1.0**
Homogeneity	**1.0**	**1.000**	**1.0**	**1.0**	**1.0**	**1.0**
Adj.Mutual Info	**1.0**	0.960	**1.0**	**1.0**	**1.0**	**1.0**

V-measure	**1.0**	0.960	**1.0**	**1.0**	**1.0**	**1.0**

As we can see in Table 3.3 the superiority of UMAP-assited clustering is confirmed when using the Calinski-Harabasz index as a way to determine the number of clusters in two out of tree algorithms. There are some variations compared to the previous approach, particularly in datasets 2 and 5, but those do not affect the overall picture.

4. Empirical illustration

4.1 Data

4.1.1 Selection and description

The initial input data of this project are time-series of user engagement metrics from Web Summit's mobile apps. However, as we will see in the following subsections, only their aggregates (sums) per event will be used for clustering.

Seven (7) user engagement metrics have been carefully selected, for each of the three (3) events we examine, based on availability, integrity and usefulness. They are of varying importance to the business and of varying relatedness to each other. This approach was chosen for two reasons: 1. behavioral patterns may vary per event/metric and 2. the essential sampling that would take place would not distort the original data as much due to their reduced size.

Each row consists of counts (positive integers), where the latter is the aggregation of user actions in the app (for the specific metric) on a daily basis. The total number of days is twenty one (21) and the conference days are included. The last day of the conference is the 14th day of the interval. The length of the interval was chosen to be such in order to avoid unnecessary sparsity and at the same time capture most user activity. Each row represents one user id. Different metrics and events do not consist necessarily of the same user ids. The time-series with just one day of activity have been been removed.

Due to confidentiality reasons, the data have been masked, but their structure has remained unchanged. The event names cannot be revealed and, thus, they will be represented as **event_a**, **event_b** and **event_c**. The days are represented as **day_1**, **day_1**,..., **day_21**. Finally, the user engagement metrics are represented as **user_engagement_metric_a**, **user_engagement_metric_b**,..., **user_engagement_metric_g**.

4.1.2 Exploration

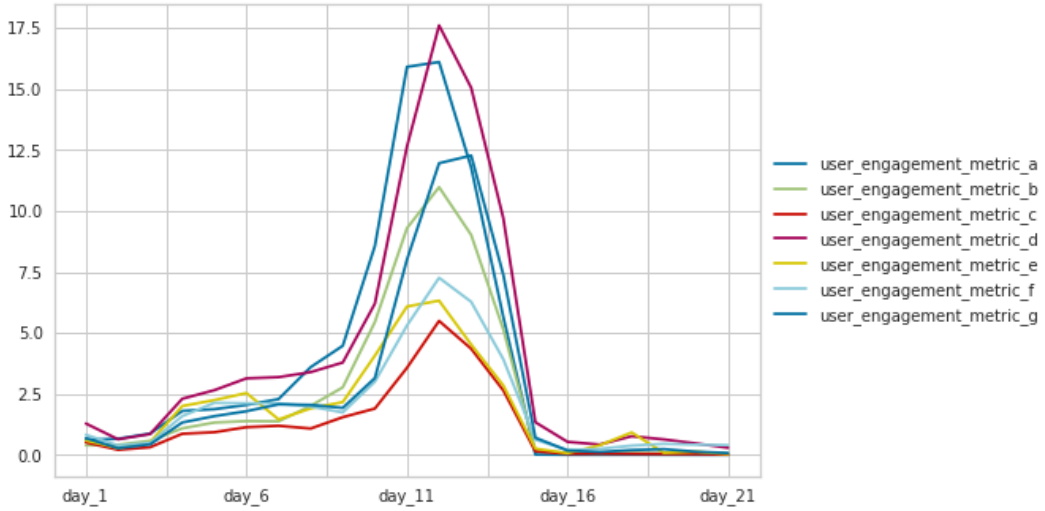
At Table 4.1, we can see an example of how the activity of five users look like. The data are quite sparse, particularly before and after the week the event is taking place. Sparsity exists in all the data and this can be confirmed by Table A.5. We can see at Figure 4.1, that for event_a the mean value of all engagement metrics increases steadily up until the event. Then, it can be observed that with some very small variations, the value decreases rapidly and by day 15, it is almost zero for

Table 4.1: Example data from user_engagement_metric_a from event_a

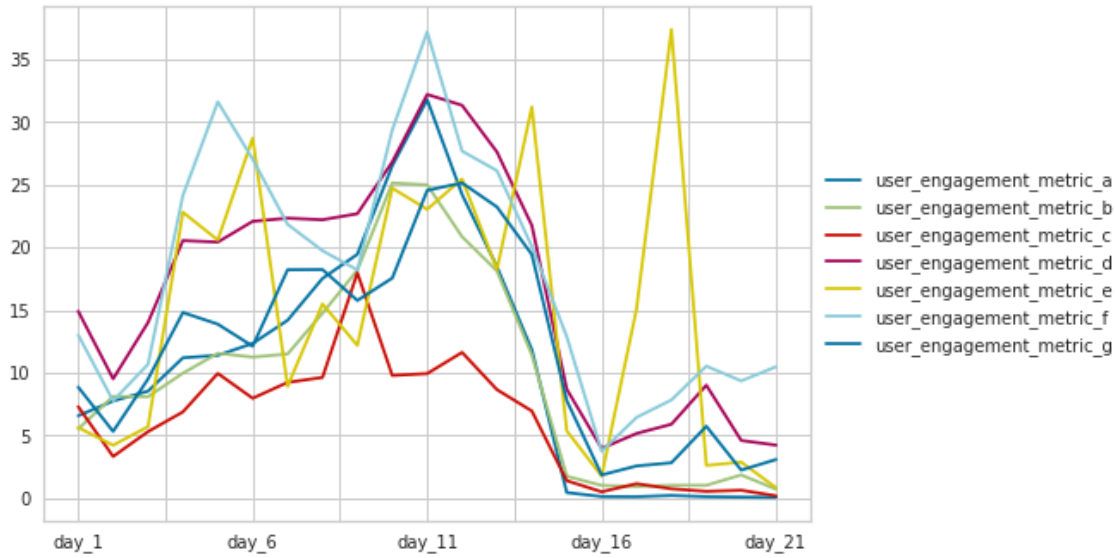
id	*day_1*	*day_2*	*day_3*	*day_4*	*day_5*	*day_6*	*day_7*	*day_8*	*day_9*	*day_10*	*day_11*	*day_12*	*day_13*	*day_14*	*day_15*	*day_16*	*day_17*	*day_18*	*day_19*	*day_20*	*day_21*
XXXXXXx	0	0	0	0	0	0	0	0	0	0	0	52	2	0	0	0	0	0	0	0	0
XXXXXXp	0	0	0	0	0	0	0	0	0	0	22	34	12	44	0	0	0	0	0	0	0
XXXXXXQ	0	0	0	0	0	0	0	0	0	0	92	14	8	0	0	0	0	0	0	0	0
XXXXXX1	0	0	0	0	0	0	0	0	0	0	34	8	40	0	0	0	0	0	0	0	0
XXXXXXP	0	0	0	0	0	0	12	0	0	80	0	0	20	4	0	0	0	0	0	0	0

most metrics. The data are obviously highly correlated and for event_a this can be seen at Table A.1. Similar conclusions can be made for the standard deviation over time, although with some noticeable variations.

Figure 4.1: Mean and standard deviation values of user activity over time for event_a



(a) Mean



(b) Standard deviation

4.2 Implementation

Due to the high correlation of user engagement metrics we choose to consider their aggregates per user per event. This would, also, facilitate some general conclusions about the overall attendee behavior. The following implementation steps are going to be taken for every aggregate:

1. **Removal of certain observations**: As mentioned in subsection 4.1.1, observations (users) with less than two days of activity are removed. The number two is chosen based on a trade-off between keeping plenty of observations for clustering and removing those observations which make behavioral patterns more difficult to interpret.

2. **Sampling**: Simple random sampling without replacement takes place in order to increase performance as much as possible. The sample size is chosen to be 5000, a number that will achieve the latter without causing unnecessary loss of information.

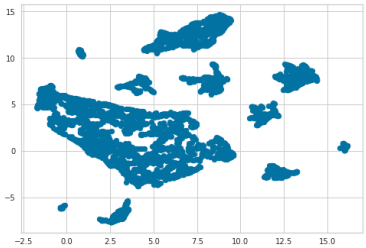
3. **Distance calculation**: Calculation of the correlation distance between all pairs of observations, since similarity in time is what we mostly care about. Another option would be the Euclidean distance on the z-score normalised time-series, but it would not be completely certain that we would get equivalent results, due to the internal mechanics of each algorithm [7].

4. **Euclidean space embedding**: Fitting of UMAP to the distance matrix. This is in accordance with chapter 3, where we showed that using UMAP can improve the clustering validation scores. In addition to that there are two more reasons: 1. HDBSCAN classifies some observations as noise. This is not always desirable when customer segmentation is the objective of clustering. As a result, the embedding to Euclidean space makes it possible to perform soft clustering and assign each noise observation to a cluster. 2. Having data in the form of (samples, features) makes the use of more internal validity indices possible.

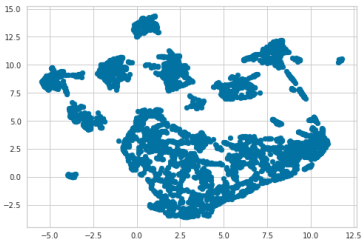
5. **Clustering**: Fitting of clustering methods to the data. The determination of the number of clusters will be done using the optimal distortion (Elbow method) and Calinski-Harabasz scores for K-Medoids and Agglomerative clustering. The Elbow method is a typical approach when it comes to this task. The Calinski-Harabasz index is a random choice among all other internal validity indices but it is frequently used in the literature. HDBSCAN handles this task on its own, after we have set certain parameters. We will set the minimum cluster size to be 50, because below that number a customer segment will not be too meaningful. We will, also, set the number of observations in a neighborhood that will make a point a core one to be 1. That is because this parameter has an effect on noise and we want the least noise possible. Finally, we will assign all noise points to their nearest clusters.

6. **Validation, comparison and selection**: Evaluation of cluster tendency with Hopkins statistic. Validation with Silhouette coefficient, Dunn index, Davies-Bouldin index and $DBCV$. Finally, we will compare the results and select the most appropriate algorithm for the segmentation.

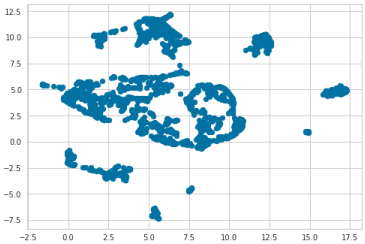
Figure 4.2: Visualization of the pairwise correlation distance of user engagement metrics using UMAP. Parameters: 1. Size of local neighborhood = 15, 2. Minimum distance between embedded points = 0.1, 3. Dimension of reduced space = 2.



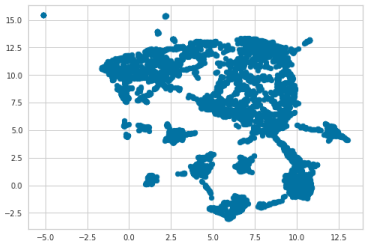
(a) event_a metric_a



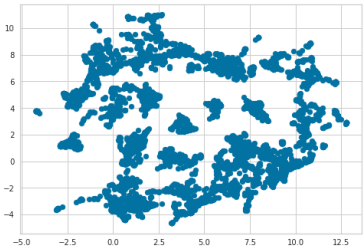
(b) event_b metric_a



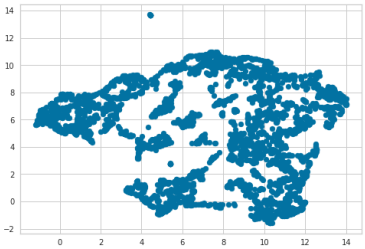
(c) event_c metric_a



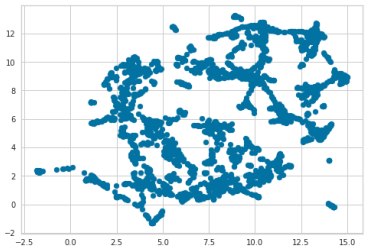
(d) event_a metric_b



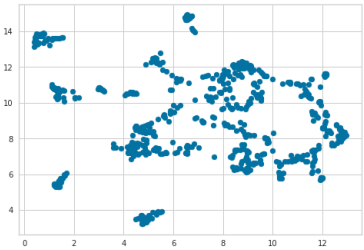
(e) event_b metric_b



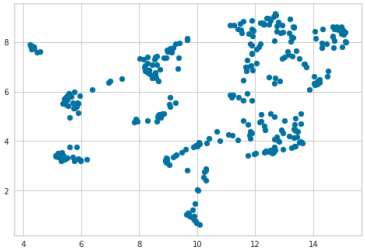
(f) event_c metric_b



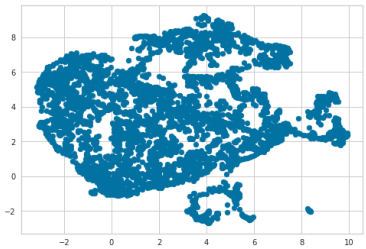
(g) event_a metric_c



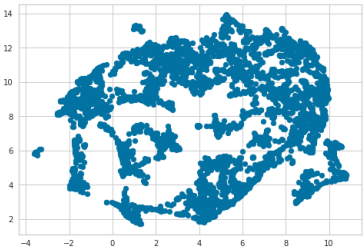
(h) event_b metric_c



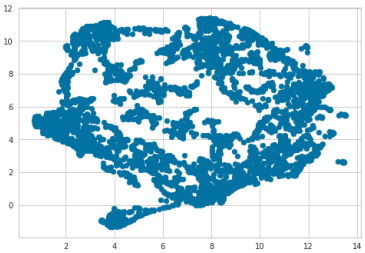
(i) event_c metric_c



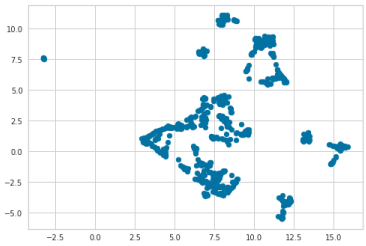
(j) event_a metric_d



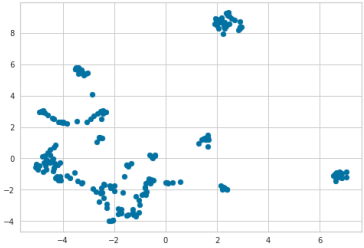
(k) event_b metric_d



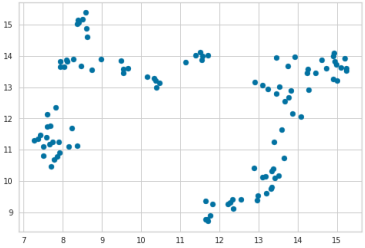
(l) event_c metric_d



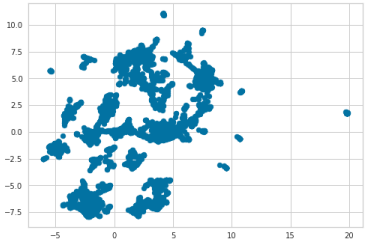
(m) event_a metric_e



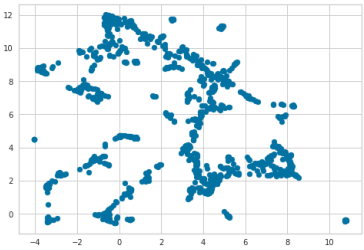
(n) event_b metric_e



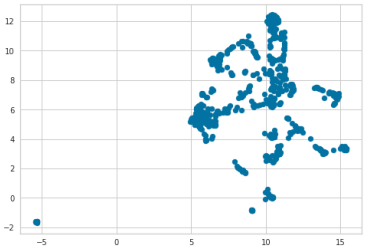
(o) event_c metric_e



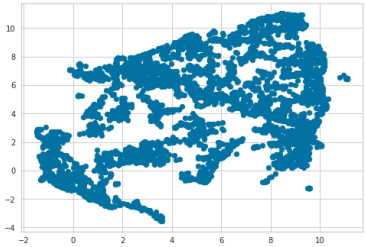
(p) event_a metric_e



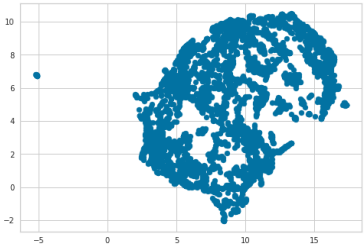
(q) event_b metric_e



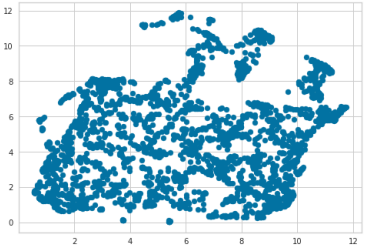
(r) event_c metric_e



(s) event_a metric_g

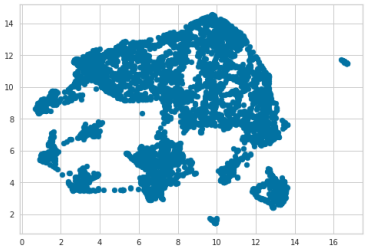


(t) event_b metric_g

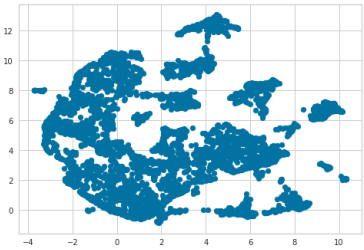


(u) event_c metric_g

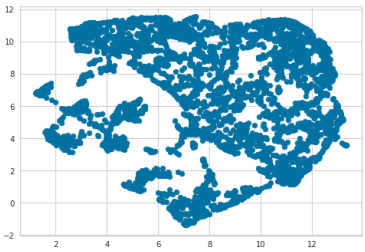
Figure 4.3: Visualization of the pairwise correlation distance of aggregates using UMAP. Parameters: 1. Size of local neighborhood = 15, 2. Minimum distance between embedded points = 0.1, 3. Dimension of reduced space = 2.



(a) event_a



(b) event_b



(c) event_c

In Figure 4.2 we can see that most metrics seem to be divided in clusters, but the degree varies. The same applies to the aggregates as seen in Figure 4.3. We will test the "clusterability" of the aggregates using the Hopkins statistic after having transformed the data with UMAP. We will use neighborhood size of 30, zero distance between embedded points and 10 dimensions of reduced space, similar to suggestions of the UMAP creators themselves [36].

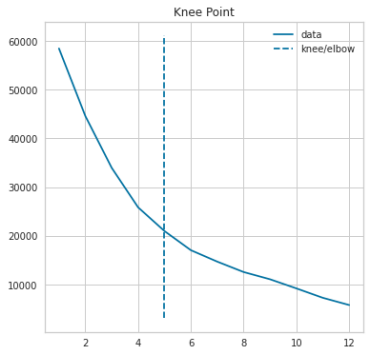
Table 4.2: Hopkins statistic on UMAP embeddings for aggregates.

event_a	event_b	event_c
0.017479	0.015997	0.020086

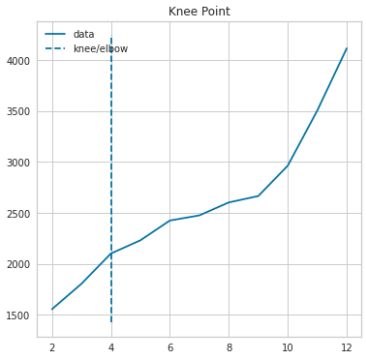
The numbers in Table 4.2 indicate that the transformed aggregates are indeed clustered and we observe high clusterability in all of them.

In Figure 4.4 and Figure 4.5 we can see the evolution of the distortion and the Calinski-Harabasz scores as the number of clusters increase. In general, we observe a tendency for the Calinski-Harabasz index to give larger numbers of clusters as the optimal ones. Moreover, we observe that in certain cases, such as Figure 4.4a, Figure 4.4b and Figure 4.4c the optimal number of clusters is not that clear. This may be either a global issue or a local issue, due to the range of numbers of clusters we chose to consider.

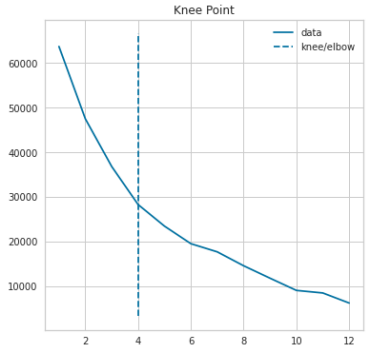
Figure 4.4: Distortion and C-H plots of the UMAP embeddings with K-medoids.



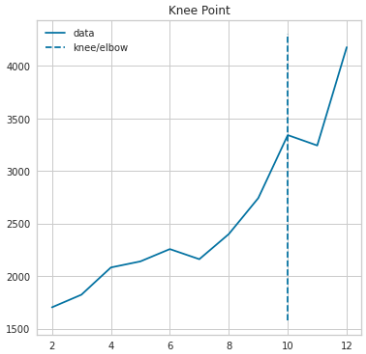
(a) event_a, distortion



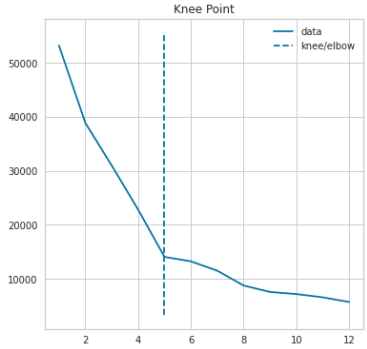
(b) event_a, C-H



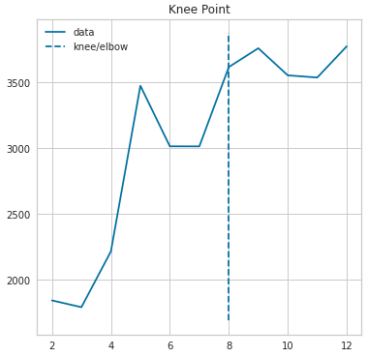
(c) event_b, distortion



(d) event_b, C-H

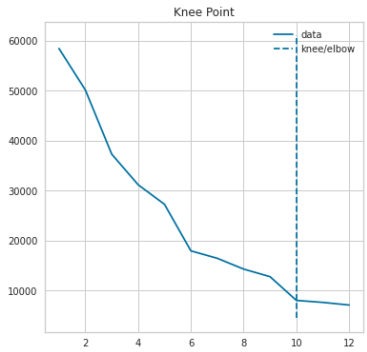


(e) event_c, distortion

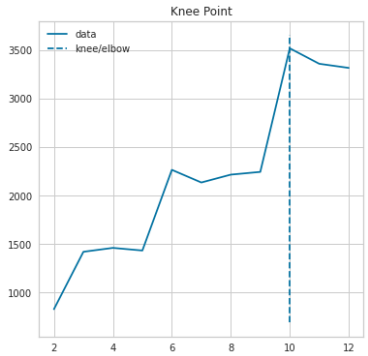


(f) event_c, C-H

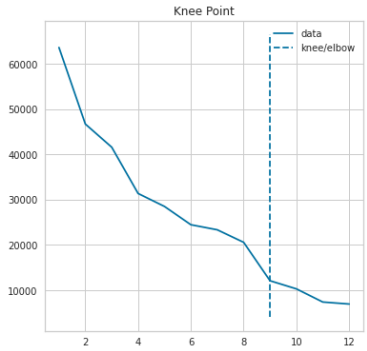
Figure 4.5: Distortion and C-H plots of the UMAP embeddings with Agglomerative clustering.



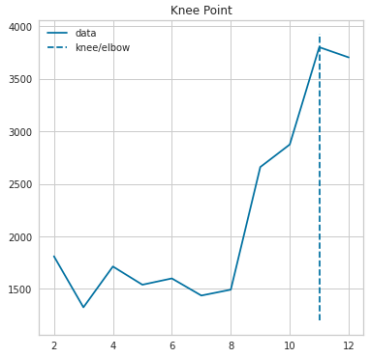
(a) event_a, distortion



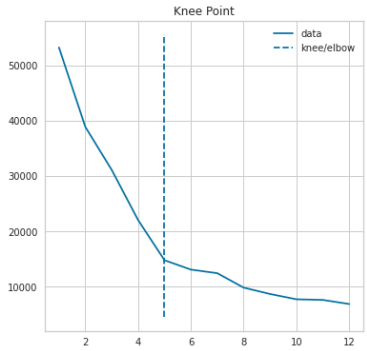
(b) event_a, C-H



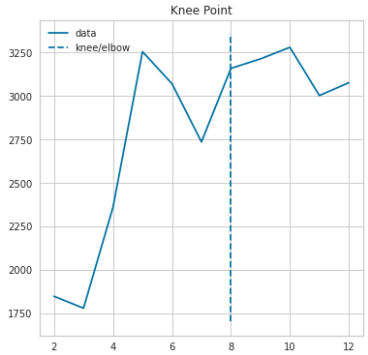
(c) event_b, distortion



(d) event_b, C-H



(e) event_c, distortion



(f) event_c, C-H

Looking at the table Table 4.3 we observe that, for event_a , Agglomerative Clustering performs better than the other methods in all indices, and the Calinski-Harabasz and distortion "agree" on the number of clusters. It is interesting that it scores better than HDBSCAN in DBCV, but this may have to do with the "soft clustering" approach that we chose for HDBSCAN.

For event_b we observe that Agglomerative Clustering (CH) and HDBSCAN perform better than K-Medoids in four and at least three out four indices respectively. Between themselves there is a "tie" because Agglomerative clustering outperforms HDBSCAN in the Silhouette coefficient and Davies-Bouldin index, while HDBSCAN has a better performance in Dunn and DBCV.

Finally, for event_c we can see that again Agglomerative Clustering (CH) performs better than HDBSCAN and K-Medoids (Dist.) in three out four indices, better than Agglomerative Clustering (Dist.) in all indices and there is a "tie" with K-Medoids (CH).

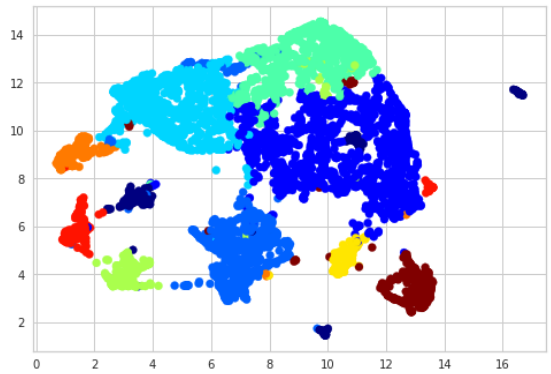
Table 4.3: Internal validity scores for aggregates.

	K-medoids (Dist.)	K-medoids (CH)	Agg.Clust (Dist.)	Agg.Clust (CH)	HDBSCAN
event_a					
Dunn	0.004	0.011	**0.031**	**0.031**	0.007
Silhouette Coef.	0.423	0.371	**0.554**	**0.554**	0.374
DBCV	-0.704	-0.714	**-0.030**	**-0.030**	-0.067
Davies-Bouldin	0.937	1.072	**0.602**	**0.602**	0.638
event_b					
Dunn	0.003	0.006	0.022	0.025	**0.094**
Silhouette Coef.	0.348	0.533	0.531	**0.579**	0.370
DBCV	-0.983	-0.277	-0.291	-0.156	**0.707**
Davies-Bouldin	1.099	0.743	0.685	**0.530**	0.661
event_c					
Dunn	0.007	0.003	0.021	**0.025**	0.009
Silhouette Coef.	0.459	**0.499**	0.441	0.452	0.157
DBCV	-0.706	-0.453	-0.532	-0.391	**0.466**
Davies-Bouldin	0.842	**0.735**	0.847	0.778	0.827

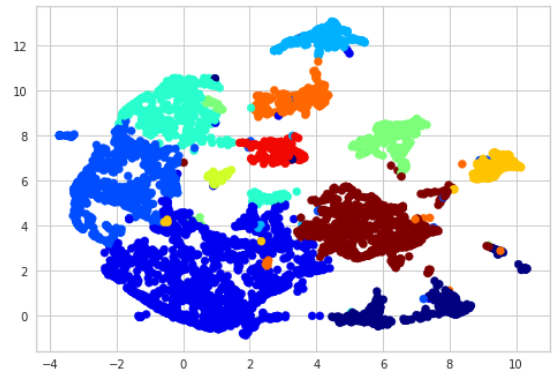
Based on the above it is reasonable to select Agglomerative Clustering (CH) for the segmentation of users. We can see in Figure 4.6 that the groupings in the

graphs are captured in a satisfying way.

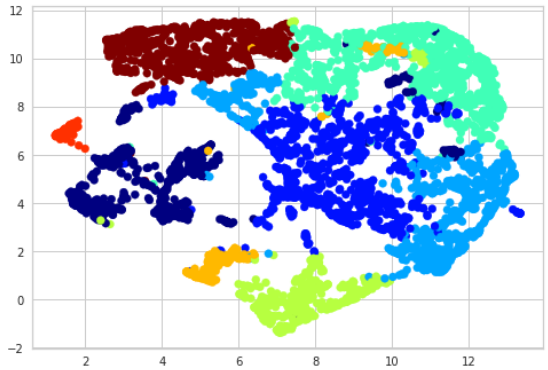
Figure 4.6: Visualization of the clustered aggregates by Agglomerative Clustering using UMAP. Parameters: 1. Size of local neighborhood = 15, 2. Minimum distance between embedded points = 0.1, 3. Dimension of reduced space = 2.



(a) event_a

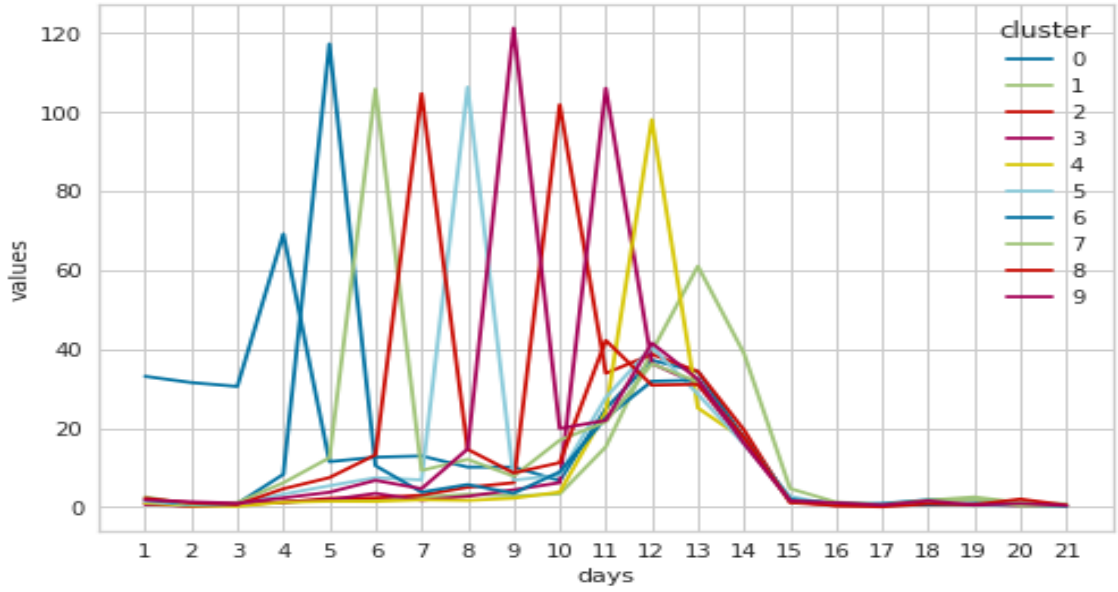


(b) event_b

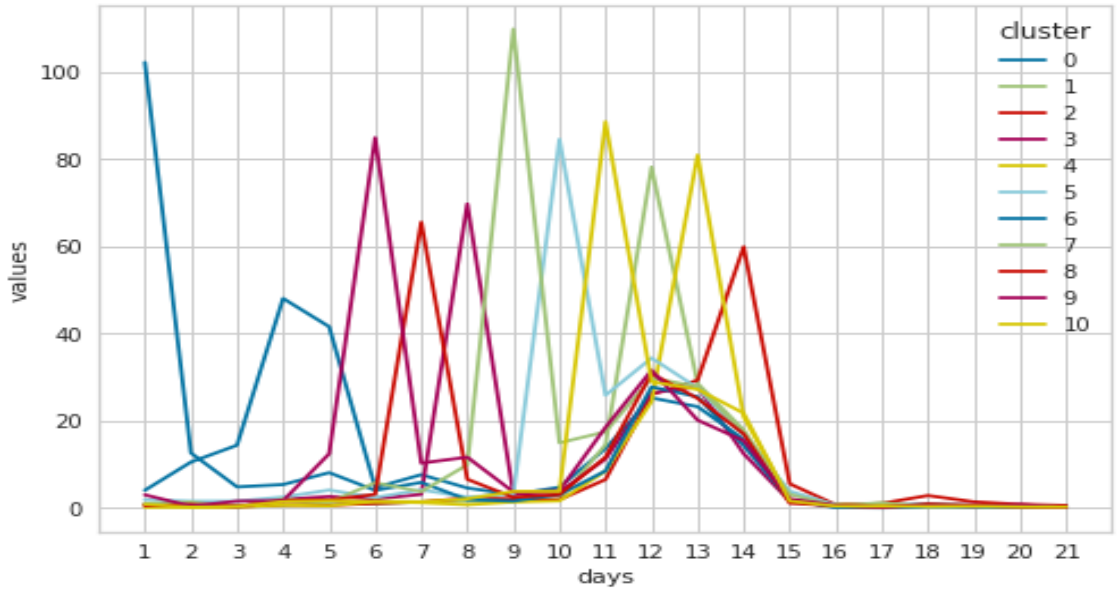


(c) event_c

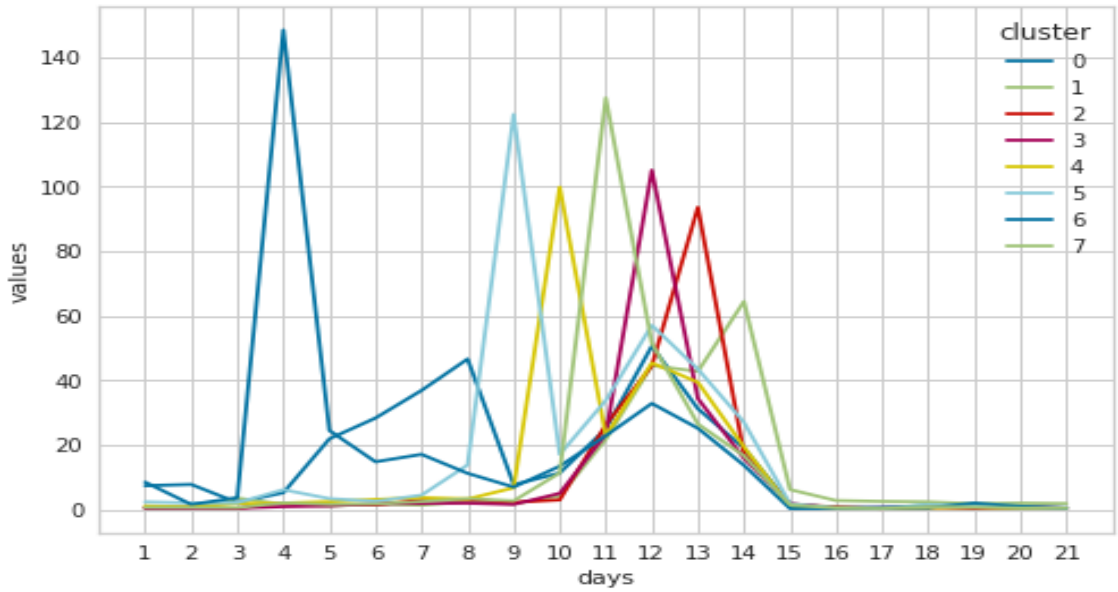
Figure 4.7: Centroids of clusters produced from Agglomerative Clustering (CH)



(a) event_a centroids



(b) event_b centroids



(c) event_c centroids

There is no straightforward way to find cluster representatives in Agglomerative clustering, as is the case for partitioning based methods. As a result, both centroids and medoids would be good candidates. At Figure 4.7, we can see that the cluster centroids in all three events have similar characteristics. Most of them have values close to zero which increase just before the event and decrease subsequently after. This trend is interrupted by spikes on different days.

5. Conclusion

The aim of this thesis was to discover segments of attendees based on their engagement with Web Summit's mobile app. In order to achieve that, we described three clustering methods each one belonging in broader groups, which represent different philosophies around clustering. We, also, described a controversial dimensionality reduction method, as well as, some assessment metrics.

We did a simulation study, generating several artificial datasets, and showed that clustering on the UMAP-transformed data performs better than clustering on the original data in several external validity indices. That effect was shown to be similar regardless of our knowledge of the number of classes in the datasets.

Finally, we worked with our data of interest. We removed from them observations that could make interpreting results difficult, we sampled them to avoid computational problems and considered their correlation distance as the most suitable measure. We transformed the data with UMAP and fitted the clustering methods. We compared the results and concluded that Agglomerative clustering with average-linkage was the most suitable one. We observed from the graphs of the cluster centroids a shared dominant pattern in all of them, However, the spikes on different days are essentially what makes them distinct.

The spikes are, precisely, what a planned intervention should be based on. Future work includes segment profiling in order to understand if these segments are being populated by attendees of varying demographic or other characteristics or if there is an underlying uniformity in those. Should the former be the case, then we can experiment with classification models (supervised learning) in order to identify to which group each new attendee belongs.

References

[1] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-Series Clustering – A Decade Review". en. In: *Information Systems* 53 (2015), pp. 16–38. ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007.

[2] Michael R. Anderberg. *Cluster Analysis for Applications*. 1st ed. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. New York: Academic Press, Inc., 1973. ISBN: 0-12-057650-3.

[3] Anthony Bagnall and Gareth Janacek. "Clustering Time Series with Clipped Data". In: *Machine Learning* 58 (2005), pp. 151–178. DOI: 10.1007/s10994-005-5825-6.

[4] A. Banerjee and R.N. Dave. "Validating clusters using the Hopkins statistic". In: *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*. Vol. 1. 2004, pp. 149–153. DOI: 10.1109/FUZZY.2004.1375706.

[5] R. Bellman and R. Kalaba. "On adaptive control processes". In: *IRE Transactions on Automatic Control* 4.2 (1959), pp. 1–9. ISSN: 1558-3651. DOI: 10.1109/TAC.1959.1104847.

[6] Benjamin Bengfort et al. *Yellowbrick*. Version 1.3.post1. Feb. 9, 2021. DOI: 10.5281/zenodo.4525724.

[7] Michael R. Berthold and Frank Höppner. *On Clustering Time Series Using Euclidean Distance and Pearson Correlation*. 2016. arXiv: 1601.02213 [cs.LG].

[8] Jack Blanchard. *POLITICO London Playbook: Bank Holiday Sizzler — Arise, Sir Ben — Boris vs. the Beeb*. POLITICO. Aug. 26, 2019. URL: https://www.politico.eu/newsletter/london-playbook/politico-london-playbook-bank-holiday-sizzler-arise-sir-ben-boris-vs-the-beeb/ (visited on 12/03/2021).

[9] T. Caliński and J Harabasz. "A dendrite method for cluster analysis". In: *Communications in Statistics* 3.1 (1974), pp. 1–27. ISSN: 0090-3272. DOI: 10.1080/03610927408827101.

[10] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 160–172. ISBN: 978-3-642-37456-2. DOI: 10.1007/978-3-642-37456-2_14.

[11] Fan Chung, Linyuan Lu, and Van Vu. "Spectra of random graphs with given expected degrees". In: *Proceedings of the National Academy of Sciences* 100.11 (2003), pp. 6313–6318. ISSN: 0027-8424. DOI: 10.1073/pnas.0937490100.

[12] H. S. M. Coxeter. *Regular Polytopes*. 1st ed. London: Methuen & Co. Ltd., 1948, pp. 120–121.

[13] David L. Davies and Donald W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[14] J. C. Dunn. "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters". In: *Journal of Cybernetics* 3.3 (1973), pp. 32–57. ISSN: 0022-0280. DOI: 10.1080/01969727308546046.

[15] J. C. Dunn†. "Well-Separated Clusters and Optimal Fuzzy Partitions". In: *Journal of Cybernetics* 4.1 (1974), pp. 95–104. ISSN: 0022-0280. DOI: 10.1080/01969727408546059.

[16] Samuel Eilenberg and J. A. Zilber. "Semi-Simplicial Complexes and Singular Homology". In: *Annals of Mathematics* 51.3 (1950), pp. 499–513. ISSN: 0003-486X. DOI: 10.2307/1969364. JSTOR: 1969364.

[17] Brian S. Everitt et al. *Cluster Analysis*. 5th ed. Wiley Series in Probability and Statistics. Chichester, UK: John Wiley & Sons, Ltd, 2011. ISBN: 978-0-470-74991-3.

[18] Evelyn Fix and J. L Hodges. *Discriminatory analysis: nonparametric discrimination, consistency properties*. Randolph Field, Tex.: USAF School of Aviation Medicine, 1951.

[19] E. B. Fowlkes and C. L. Mallows. "A Method for Comparing Two Hierarchical Clusterings". In: *Journal of the American Statistical Association* 78.383 (1983), pp. 553–569. DOI: 10.1080/01621459.1983.10478008.

[20] Ilan Gronau and Shlomo Moran. "Optimal Implementations of UPGMA and Other Common Clustering Algorithms". In: *Information Processing Letters* 104.6 (2007), pp. 205–210. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2007.07.002.

[21] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 3rd ed. The Morgan Kaufmann Series in Data Management Systems. Waltham, MA: Morgan Kaufmann Publishers, 2012. ISBN: 978-0-12-381479-1.

[22] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[23] Brian Hopkins and J. G. Skellam. "A New Method for determining the Type of Distribution of Plant Individuals". In: *Annals of Botany* 18.2 (Apr. 1954), pp. 213–227. ISSN: 0305-7364. DOI: 10.1093/oxfordjournals.aob.a083391.

[24] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/bf01908075.

[25] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[26] Brijnesh J. Jain and David Schultz. *Optimal Warping Paths are unique for almost every Pair of Time Series*. 2018. arXiv: 1705.05681 [cs.LG].

[27] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. 1st ed. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, Inc., 1990, 2005. ISBN: 0-47-1-73578-7.

[28] Leonard Kaufmann and Peter Rousseeuw. "Clustering by Means of Medoids". In: *Statistical Data Analysis based on the L1-Norm and Related Methods* (1987), pp. 405–416.

[29] Philip Kotler and Kevin Lane Keller. *Marketing Management*. 12th ed. Upper Saddle River, NJ: Prentice Hall, 2006. ISBN: 0-13-145757-8.

[30] Saunders Mac Lane. *Categories for the working mathematician*. Vol. 5. Springer Science & Business Media, 2013. ISBN: 978-1-4757-4721-8.

[31] Connected Intelligence Limited. Web Summit Group. URL: https://websummit.com/ (visited on 08/21/2021).

[32] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. 1st ed. New York, NY: Springer Science+Business Media, Inc., 2005. ISBN: 978-0-387-24435-8.

[33] Peter J. May. *Simplicial objects in algebraic topology*. University of Chicago Press, 1992. 171 pp. ISBN: 978-0-226-51181-8.

[34] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *The Journal of Open Source Software* 2.11 (2017), p. 205.

[35] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: (2020). arXiv: 1802.03426 [stat.ML].

[36] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29 (2018), p. 861.

[37] Michael McKerns and Michael Aivazis. *pathos: a framework for heterogeneous computing*. 2010.

[38] Michael M. McKerns et al. "Building a Framework for Predictive Science". In: *Proceedings of the 10th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2011, pp. 76–86. DOI: 10.25080/Majora-ebaa42b7-00d.

[39] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[40] Davoud Moulavi et al. "Density-Based Clustering Validation". In: *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, 2014, pp. 839–847. DOI: 10.1137/1.9781611973440.96.

[41] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. "Otakar Borůvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history". In: *Discrete Mathematics*. Czech and Slovak 2 233.1 (2001), pp. 3–36. ISSN: 0012-365X. DOI: 10.1016/S0012-365X(00)00224-7.

[42] Hae-Sang Park and Chi-Hyuck Jun. "A simple and fast algorithm for K-Medoids clustering". In: *Expert Systems with Applications* 36.2, Part 2 (2009), pp. 3336–3341. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2008.01.039.

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

[44] Eréndira Rendón et al. "Internal versus External cluster validation indexes". In: *International Journal of computers and communications* 5.1 (2011), pp. 27–34.

[45] Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017. ISBN: 978-0-486-82080-4.

[46] Andrew Rosenberg and Julia Hirschberg. "V-Measure: A conditional entropy-based external cluster evaluation measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 410–420. DOI: 10.7916/D80V8N84.

[47] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: 10.1016/0377-0427(87)90125-7.

[48] Ville Satopaa et al. "Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior". In: *2011 31st International Conference on Distributed Computing Systems Workshops*. June 2011, pp. 166–171. DOI: 10.1109/ICDCSW.2011.20.

[49] Robert R. Sokal and Charles Duncan Michener. "A statistical method for evaluating systematic relationships". In: *University of Kansas science bulletin* 38 (1958), pp. 1409–1438.

[50] The pandas development team. *pandas-dev/pandas: Pandas*. Version 1.2.5. June 2021. DOI: 10.5281/zenodo.5013202.

[51] Konstantinos Tsiptsis and Antonios Chorianopoulos. *Data Mining Techniques in CRM. Inside Customer Segmentation*. 1st ed. Chichester, UK: John Wiley & Sons, Ltd, 2009. ISBN: 9780470743973.

[52] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance". In: *Journal of Machine Learning Research* 11.95 (2010), pp. 2837–2854.

[53] T. Warren Liao. "Clustering of Time Series Data—a Survey". In: *Pattern Recognition* 38.11 (2005), pp. 1857–1874. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2005.01.025.

Appendices

A. Primary data: summary statistics and plots

Table A.1: Correlation matrix of mean values of event_a

	user_engagement_metric_a	user_engagement_metric_b	user_engagement_metric_c	user_engagement_metric_d	user_engagement_metric_e	user_engagement_metric_f	user_engagement_metric_g
user_engagement_metric_a	1.000	0.989	0.955	0.955	0.965	0.950	0.913
user_engagement_metric_b	0.989	1.000	0.980	0.985	0.955	0.974	0.959
user_engagement_metric_c	0.955	0.980	1.000	0.993	0.939	0.992	0.977
user_engagement_metric_d	0.955	0.985	0.993	1.000	0.928	0.989	0.990
user_engagement_metric_e	0.965	0.955	0.939	0.928	1.000	0.950	0.879
user_engagement_metric_f	0.950	0.974	0.992	0.989	0.950	1.000	0.976
user_engagement_metric_g	0.913	0.959	0.977	0.990	0.879	0.976	1.000

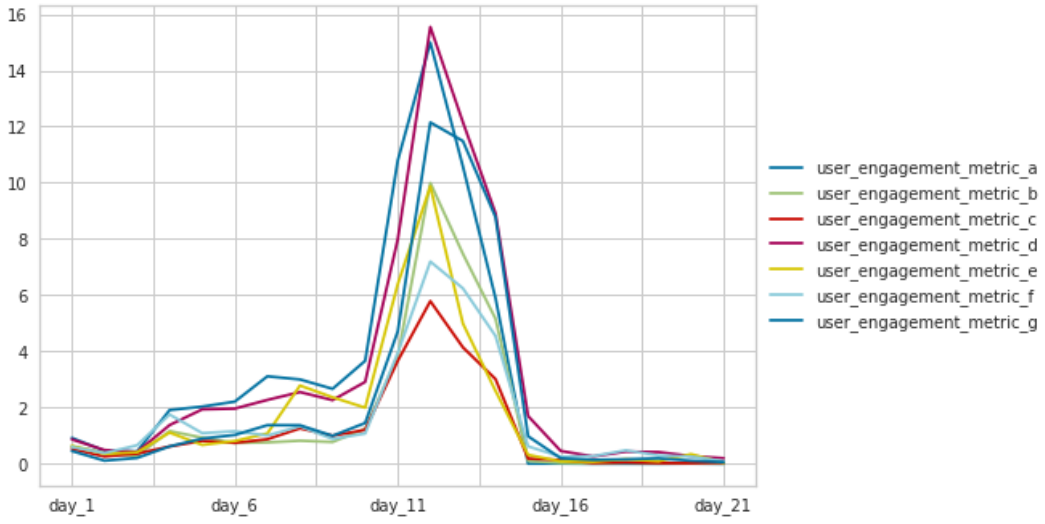
Table A.2: Correlation matrix of mean values of event_b

	user_engagement_metric_a	*user_engagement_metric_b*	*user_engagement_metric_c*	*user_engagement_metric_d*	*user_engagement_metric_e*	*user_engagement_metric_f*	*user_engagement_metric_g*
user_engagement_metric_a	1.000	0.945	0.989	0.966	0.974	0.953	0.908
user_engagement_metric_b	0.945	1.000	0.974	0.990	0.906	0.988	0.983
user_engagement_metric_c	0.989	0.974	1.000	0.988	0.961	0.978	0.950
user_engagement_metric_d	0.966	0.990	0.988	1.000	0.925	0.989	0.983
user_engagement_metric_e	0.974	0.906	0.961	0.925	1.000	0.902	0.847
user_engagement_metric_f	0.953	0.988	0.978	0.989	0.902	1.000	0.981
user_engagement_metric_g	0.908	0.983	0.950	0.983	0.847	0.981	1.000

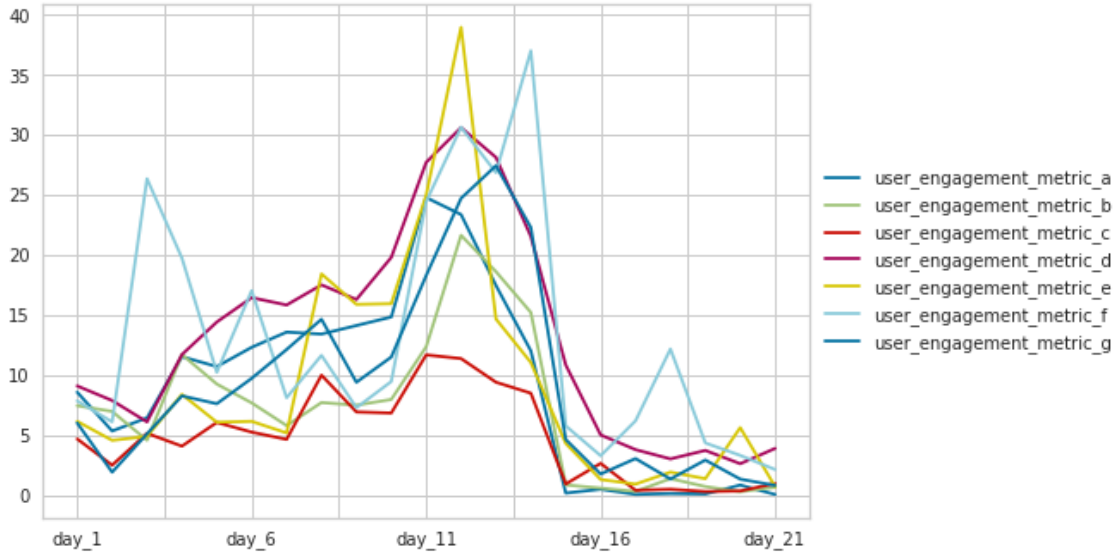
Table A.3: Correlation matrix of mean values of event_c

	user_engagement_metric_a	*user_engagement_metric_b*	*user_engagement_metric_c*	*user_engagement_metric_d*	*user_engagement_metric_e*	*user_engagement_metric_f*	*user_engagement_metric_g*
user_engagement_metric_a	1.000	0.946	0.969	0.966	0.988	0.952	0.927
user_engagement_metric_b	0.946	1.000	0.940	0.993	0.904	0.972	0.986
user_engagement_metric_c	0.969	0.940	1.000	0.949	0.951	0.951	0.902
user_engagement_metric_d	0.966	0.993	0.949	1.000	0.931	0.982	0.991
user_engagement_metric_e	0.988	0.904	0.951	0.931	1.000	0.920	0.885
user_engagement_metric_f	0.952	0.972	0.951	0.982	0.920	1.000	0.970
user_engagement_metric_g	0.927	0.986	0.902	0.991	0.885	0.970	1.000

Figure A.1: Mean and standard deviation values of user activity over time for event_b

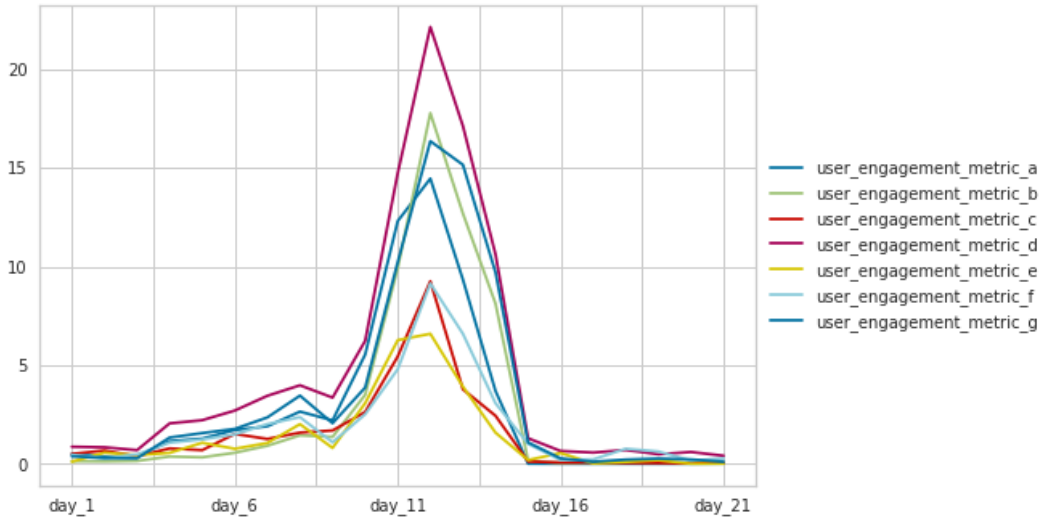


(a) Mean

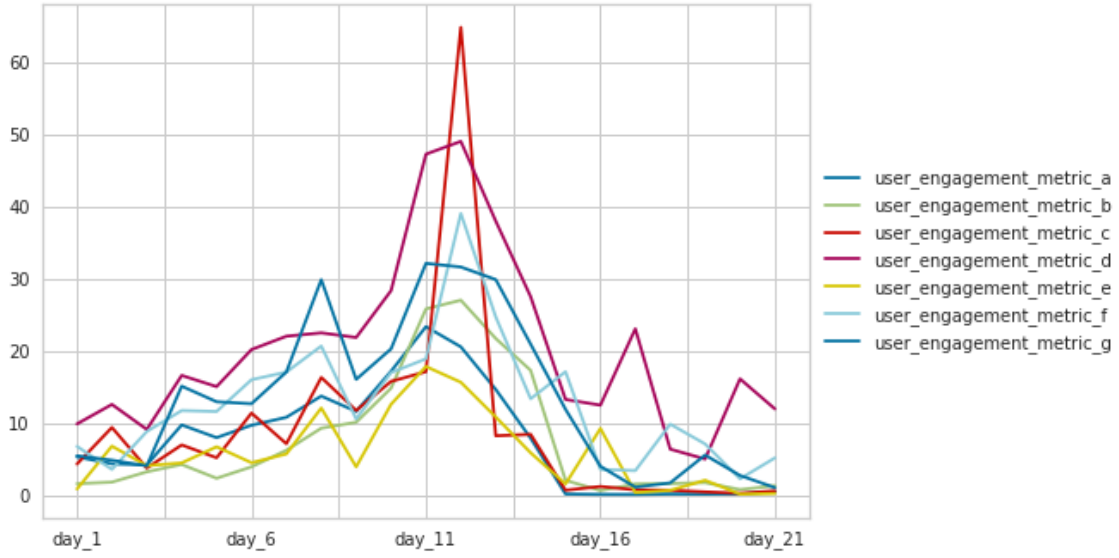


(b) Standard deviation

Figure A.2: Mean and standard deviation values of user activity over time for event_c



(a) Mean



(b) Standard deviation

Table A.4: Summary of user_engagement_metric_a from event_a

	day_1	day_2	day_3	day_4	day_5	day_6	day_7
count	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000
mean	0.619	0.659	0.873	1.802	1.874	2.047	2.287
std	6.572	7.748	8.536	11.196	11.391	12.341	14.189
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
50%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
75%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
max	270.000	426.000	292.000	322.000	604.000	438.000	708.000

	day_8	day_9	day_10	day_11	day_12	day_13	day_14
count	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000
mean	3.597	4.472	8.559	15.905	16.104	11.779	5.659
std	17.524	19.451	26.510	31.782	24.216	18.461	11.841
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
50%	0.000	0.000	0.000	2.000	6.000	4.000	0.000
75%	0.000	0.000	2.000	18.000	22.000	16.000	6.000
max	474.000	536.000	778.000	844.000	366.000	364.000	470.000

	day_15	day_16	day_17	day_18	day_19	day_20	day_21
count	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000	23852.000
mean	0.014	0.005	0.004	0.006	0.004	0.002	0.001
std	0.451	0.130	0.116	0.220	0.119	0.085	0.067
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
50%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
75%	0.000	0.000	0.000	0.000	0.000	0.000	0.000
max	50.000	8.000	10.000	26.000	8.000	8.000	8.000

Table A.5: Average sparsity of user engagement metrics

	event_a	event_b	event_c
user_engagement_metric_a	0.842896	0.856351	0.857400
user_engagement_metric_b	0.834639	0.859882	0.841713
user_engagement_metric_c	0.857328	0.865202	0.861117
user_engagement_metric_d	0.775313	0.810510	0.789079
user_engagement_metric_e	0.884396	0.887041	0.886952
user_engagement_metric_f	0.866077	0.875184	0.869774
user_engagement_metric_g	0.818803	0.834133	0.812528

B. Simulation data: evaluation

Table B.1: External validity scores for z-score normalized artificial datasets 1-10 with known clusters

	UMAP+K-med.	K-med.	UMAP+Agg.Cl.	Agg.Cl.	UMAP+HDB.	HDB.
Dataset 1						
Adj.Rand	**0.903**	0.655	0.892	0.735	0.792	0.820
Completeness	**0.811**	0.564	0.789	0.752	0.803	0.694
Fowlkes–Mallows	**0.945**	0.797	0.938	0.862	0.890	0.897
Homogeneity	**0.814**	0.649	0.813	0.577	0.661	0.741
Adj.Mutual Info	**0.812**	0.603	0.801	0.653	0.725	0.717
V-measure	**0.812**	0.604	0.801	0.653	0.725	0.717
Dataset 2						
Adj.Rand	**0.597**	0.411	0.567	0.001	-0.001	0.508
Completeness	**0.485**	0.317	0.464	0.053	0.004	0.402
Fowlkes–Mallows	**0.802**	0.709	0.789	0.711	0.704	0.758
Homogeneity	**0.484**	0.319	0.456	0.001	0.000	0.401
Adj.Mutual Info	**0.484**	0.318	0.460	0.002	0.000	0.402
V-measure	**0.485**	0.318	0.460	0.002	0.001	0.402
Dataset 3						
Adj.Rand	**0.998**	0.996	**0.998**	0.993	**0.998**	0.988
Completeness	**0.994**	0.988	**0.994**	0.982	**0.994**	0.973
Fowlkes–Mallows	**0.999**	0.998	**0.999**	0.997	**0.999**	0.994
Homogeneity	**0.994**	0.989	**0.994**	0.982	**0.994**	0.973
Adj.Mutual Info	**0.994**	0.989	**0.994**	0.982	**0.994**	0.973
V-measure	**0.994**	0.989	**0.994**	0.982	**0.994**	0.973
Dataset 4						
Adj.Rand	**0.997**	0.860	**0.997**	0.983	**0.997**	0.995
Completeness	**0.995**	0.918	**0.995**	0.992	**0.995**	0.990
Fowlkes–Mallows	**0.998**	0.881	**0.998**	0.985	**0.998**	0.996
Homogeneity	**0.995**	0.963	**0.995**	0.969	**0.995**	0.990
Adj.Mutual Info	**0.995**	0.940	**0.995**	0.980	**0.995**	0.990

V-measure	**0.995**	0.940	**0.995**	0.980	**0.995**	0.990
Dataset 5						
Adj.Rand	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Completeness	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Fowlkes–Mallows	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Homogeneity	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Adj.Mutual Info	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
V-measure	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Dataset 6						
Adj.Rand	**0.994**	0.990	**0.994**	0.886	**0.994**	0.984
Completeness	**0.988**	0.980	**0.988**	0.980	**0.988**	0.970
Fowlkes–Mallows	**0.995**	0.992	**0.995**	0.916	**0.995**	0.987
Homogeneity	**0.988**	0.981	**0.988**	0.869	**0.988**	0.975
Adj.Mutual Info	**0.988**	0.980	**0.988**	0.922	**0.988**	0.972
V-measure	**0.988**	0.980	**0.988**	0.922	**0.988**	0.972
Dataset 7						
Adj.Rand	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Completeness	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Fowlkes–Mallows	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Homogeneity	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Adj.Mutual Info	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
V-measure	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Dataset 8						
Adj.Rand	**0.997**	0.816	**0.997**	0.948	**0.997**	0.996
Completeness	**0.994**	0.870	**0.994**	0.980	**0.994**	0.991
Fowlkes–Mallows	**0.998**	0.852	**0.998**	0.958	**0.998**	0.996
Homogeneity	**0.993**	0.940	**0.993**	0.926	**0.993**	0.991
Adj.Mutual Info	**0.994**	0.903	**0.994**	0.952	**0.994**	0.991
V-measure	**0.994**	0.904	**0.994**	0.953	**0.994**	0.991
Dataset 9						
Adj.Rand	0.488	0.366	**0.747**	0.549	0.513	0.443
Completeness	0.540	0.439	**0.647**	0.540	0.622	0.459
Fowlkes–Mallows	0.589	0.485	**0.806**	0.671	0.649	0.552
Homogeneity	**0.616**	0.529	0.605	0.442	0.464	0.477
Adj.Mutual Info	0.573	0.478	**0.623**	0.483	0.529	0.466
V-measure	0.575	0.480	**0.625**	0.486	0.531	0.468
Dataset 10						
Adj.Rand	0.582	0.576	**0.979**	0.958	**0.979**	0.791
Completeness	0.592	0.572	**0.952**	0.910	**0.952**	0.693
Fowlkes–Mallows	0.766	0.762	**0.990**	0.979	**0.990**	0.894

Homogeneity	0.838	0.811	**0.944**	0.905	**0.944**	0.803
Adj.Mutual Info	0.694	0.671	**0.948**	0.907	**0.948**	0.744
V-measure	0.694	0.671	**0.948**	0.907	**0.948**	0.744

C. UMAP-transformed data: example tables

Table C.1: Random sample of UMAP-transformed event_a aggregates in 10 dimensions.

1	2	3	4	5	6	7	8	9	10
2.2458	1.5888	9.2910	7.7264	6.3383	6.5498	2.6738	1.7395	7.6562	3.9353
2.6537	0.8798	5.6589	9.2574	6.5118	8.0717	4.3155	3.3288	6.4362	4.6587
2.1628	4.4724	5.5382	6.3201	4.3639	8.4362	2.9336	3.7194	7.4005	3.7475
3.3124	0.6617	5.4705	9.6279	6.4506	8.3380	4.2881	3.8196	6.3632	4.5680
3.4921	4.1638	3.6812	7.9891	5.8691	8.0641	3.7711	3.4985	7.1981	4.4459
2.8628	1.1507	5.3400	7.9826	6.6347	7.4409	4.0828	2.4086	6.7561	4.6505
4.8913	3.4882	4.4138	8.6739	6.1623	8.0052	3.8874	3.4654	7.0676	4.5825
4.5956	0.8022	4.5180	7.8837	6.6674	7.5892	3.9740	2.7787	6.7397	4.5146
1.7047	1.2442	7.5191	5.2302	8.1975	7.0322	4.7455	1.5215	6.3284	4.8477
2.5576	1.5940	9.3015	7.6172	6.3760	6.5792	2.4431	1.7433	7.8323	3.7423
4.0574	0.8354	4.3555	7.9070	6.6427	7.6623	4.0498	2.8125	6.6986	4.5444
1.9816	4.6986	5.6649	6.0685	4.1590	8.4650	2.8333	3.7691	7.4527	3.6663
1.6867	0.8036	7.9704	5.6509	6.2770	4.6028	3.6356	1.3989	7.4223	4.6406
1.2650	2.3423	4.3000	4.6939	7.6168	7.9917	4.3510	2.6107	6.9441	4.5949
1.2900	1.4964	7.2021	8.3482	6.7993	7.3348	4.2201	1.9543	6.7072	4.8248
4.0379	4.1208	4.6357	8.7254	6.0497	7.9553	3.8910	3.3617	7.1598	4.6372
1.0156	1.3007	5.9544	8.5378	6.5908	7.7032	4.3106	2.5717	6.5896	4.7438
4.6055	4.8559	3.8468	8.6131	5.8657	8.1309	3.7711	3.7013	7.2896	4.5411
1.3702	1.1069	6.6056	9.3919	6.5383	7.9613	4.4069	3.0047	6.4169	4.7674
2.2846	1.1812	7.2429	5.2931	8.0936	7.0556	4.6308	1.5749	6.4021	4.7804
1.7201	0.9646	6.8354	9.9511	6.4865	8.1665	4.4539	3.3786	6.3067	4.7599
0.6924	1.3571	3.3405	7.1794	6.5029	7.8784	4.1299	2.8662	6.7324	4.4695
1.4047	1.4140	7.9052	8.9119	6.7752	7.3340	4.2569	2.0148	6.6423	4.8724
0.7686	1.3811	5.2582	7.9373	6.6141	7.6592	4.2475	2.4473	6.6745	4.6800
5.9468	0.5992	4.7611	8.1645	6.8064	7.5985	4.0687	2.7631	6.6628	4.6014
1.4947	0.8837	5.2450	9.3589	6.4067	8.3427	4.4178	3.6669	6.3416	4.6284
4.2344	0.8450	3.8090	7.7325	6.5939	7.7630	3.9911	3.0090	6.7216	4.4610
2.0190	1.2247	4.5740	7.6070	6.6009	7.5632	4.0970	2.5070	6.7617	4.5906
4.8193	5.3345	3.7441	8.5612	5.7650	8.0584	3.6759	3.6706	7.4235	4.5256

1.6575	1.1447	2.8080	7.1437	6.4796	7.8833	4.0561	3.0811	6.7362	4.4033
0.8042	1.4094	3.8400	7.2536	6.5474	7.7670	4.1429	2.6746	6.7493	4.5316
5.9584	0.5531	4.2674	8.0865	6.7691	7.7378	4.0234	3.0294	6.6559	4.5137
1.0030	2.1721	4.1857	5.4691	7.2979	7.9771	4.2880	2.6819	6.8574	4.6015
1.3283	0.9061	4.4189	8.9820	6.3651	8.4176	4.3687	3.7702	6.3832	4.5443
3.9333	1.0270	6.0768	8.7439	6.6815	7.6000	4.1755	2.6708	6.6218	4.7135
1.1118	1.4259	3.2843	7.5491	6.4523	8.1094	4.1607	3.2562	6.6761	4.4549
2.0687	1.3363	5.7901	7.9710	6.6515	7.3667	4.1086	2.1854	6.7787	4.6978
5.7250	0.5787	5.0784	8.4072	6.7245	7.6995	3.9303	3.0849	6.6939	4.4683
6.3651	0.6284	4.9552	8.0386	6.6439	7.6590	3.7351	3.1318	6.7918	4.3398
1.6575	1.3638	8.2838	9.0918	6.8175	7.3506	4.2571	2.0613	6.6071	4.8743
5.1830	5.4684	3.9570	8.5786	5.6294	8.0985	3.6039	3.7475	7.4527	4.4791
4.8095	1.1376	6.7919	6.3787	6.3964	6.3604	2.1467	2.4322	7.8398	3.4672
0.7899	1.4218	5.5657	7.9902	6.6319	7.5728	4.2391	2.3200	6.6947	4.7117
4.8714	0.5882	4.8601	8.7647	6.6264	7.9775	4.1070	3.3761	6.5461	4.5213
1.6751	0.8133	7.8779	5.6462	6.2270	4.5962	3.6342	1.4249	7.4310	4.6469
2.5845	0.9340	6.1459	9.3981	6.5366	7.9949	4.3368	3.1957	6.4316	4.7114
5.6351	0.6626	5.4839	8.5254	6.7762	7.6129	4.0825	2.8176	6.6403	4.6244
1.8893	1.1274	3.0575	7.2038	6.5125	7.8321	4.0528	2.9876	6.7434	4.4267
5.5623	0.6521	4.1717	8.0494	6.7188	7.7541	4.0156	3.0510	6.6704	4.5036
4.3786	4.5390	4.6130	8.7857	5.9783	7.9181	3.8333	3.3681	7.2659	4.6547

Table C.2: Random sample of UMAP-transformed event_b aggregates in 10 dimensions.

1	2	3	4	5	6	7	8	9	10
1.7046	6.0762	1.8502	6.4533	7.8263	6.7182	5.1871	6.4045	4.6159	7.2118
2.5369	5.8501	3.4897	5.0636	8.6582	3.4834	4.6363	6.4494	4.4965	7.2641
6.1053	4.9654	1.5794	5.7908	6.1653	6.4625	5.2901	6.5189	4.2521	7.2954
2.8376	4.0451	1.9985	6.3479	6.9701	6.5937	5.2185	6.2982	4.3158	7.2667
4.1486	4.2112	3.5854	3.6091	3.9830	6.2378	5.7514	6.5827	4.4308	8.4538
1.0864	2.0685	2.9172	6.8203	6.4699	6.5026	5.0079	5.9477	4.1351	7.3894
1.2157	7.6224	1.7416	6.4873	8.1335	6.7159	5.1882	6.4600	4.8074	7.1925
0.8082	4.1794	1.5775	5.9904	5.8065	5.2502	5.1194	5.7599	4.4415	7.4980
0.6092	4.3776	1.8807	6.3951	7.0288	6.1071	5.1221	6.0292	4.4903	7.3402
0.8310	6.9150	1.0835	6.0474	6.2086	5.5963	5.2426	5.8996	4.7875	7.4754
0.7042	6.2154	1.5307	6.3258	7.2629	6.1512	5.2094	6.1187	4.7201	7.3349
1.3880	5.5931	4.4887	10.0466	6.0089	7.3789	4.9549	3.9207	3.4007	6.9192
2.3891	6.0128	3.3089	5.1235	8.5339	3.6430	4.6983	6.4274	4.5444	7.2771
5.1390	3.9004	1.7706	6.0295	6.3935	6.5181	5.2192	6.4410	4.1587	7.2474

1.2125	7.8862	1.4617	6.2624	7.4760	6.3801	5.2142	6.3084	4.8323	7.2876
0.6127	2.4299	2.4337	6.5813	6.6239	6.1994	4.9649	5.9537	4.1855	7.3459
0.8258	7.8494	1.2256	6.1943	7.0524	6.0766	5.2554	6.1399	4.8893	7.3802
3.9169	4.5050	4.3014	9.5293	5.6969	7.5362	4.0510	5.1333	3.6864	7.0347
0.8947	6.3325	1.5315	6.3251	7.3138	6.2206	5.2194	6.1636	4.7148	7.3186
6.0740	4.9334	1.5683	5.8098	6.1231	6.4396	5.2879	6.4852	4.2311	7.2859
3.9323	4.8249	1.4820	6.0134	6.4735	6.3364	5.2989	6.3153	4.3425	7.2888
2.4631	5.9597	3.2218	5.1458	8.3699	3.7292	4.7277	6.4004	4.5208	7.2879
4.9011	4.3583	5.1045	9.3448	5.0717	7.9667	2.7090	6.3138	4.2909	7.5038
0.7376	2.0438	2.5903	6.6706	6.5187	6.2567	4.9275	5.9352	4.1146	7.3415
5.5846	4.8491	2.2330	5.2087	5.8267	6.5048	5.4119	6.6023	4.3364	7.5815
0.5972	2.5460	2.3161	6.4931	6.4765	6.0445	4.9871	5.9087	4.2049	7.3693
0.3000	7.5238	1.0655	5.8793	4.5650	5.4395	4.9549	5.6812	4.7368	7.4477
1.1978	1.6334	2.7604	6.7037	6.1532	6.2015	4.9368	5.8720	4.0216	7.3811
1.2681	3.8962	5.5046	8.8057	8.4697	8.4406	5.6551	5.9007	4.5737	7.5570
3.2340	5.5988	3.4265	5.1555	8.3308	3.8347	4.6532	6.4854	4.4310	7.2454
3.6676	3.2128	1.9693	6.2647	6.4073	6.4212	5.1272	6.2601	4.1275	7.2517
0.8619	8.0077	1.1507	6.1694	6.7912	5.9786	5.2671	6.0544	4.9025	7.4221
1.0179	2.3014	2.1181	6.2280	5.5785	5.5272	5.0057	5.7187	4.1396	7.4688
2.3658	5.7220	3.2393	5.2705	8.3681	3.9139	4.7163	6.3852	4.4945	7.2692
0.7808	7.1733	1.4205	6.3002	7.3387	6.1989	5.2248	6.1765	4.8096	7.3270
1.6453	7.7310	1.6628	6.4229	8.0039	6.7528	5.2041	6.4976	4.7939	7.1965
1.0270	2.3722	2.0781	6.1959	5.5423	5.4930	5.0188	5.7052	4.1440	7.4778
4.1506	5.1255	3.7719	7.5005	5.9054	6.1730	2.1146	6.8438	4.9459	7.1922
0.7567	4.9540	1.7889	6.3969	7.2084	6.2135	5.1495	6.1014	4.5722	7.3231
4.3069	4.3519	3.6484	3.4889	3.9963	6.2198	5.7504	6.6254	4.4738	8.4950
1.2932	6.3955	0.8076	5.5000	4.2969	5.3796	5.3394	5.5947	4.4500	7.5992
5.4695	5.2355	1.6350	5.9053	6.7362	6.7089	5.3762	6.5958	4.3328	7.2463
2.1455	6.0519	3.1074	5.3043	8.4303	4.0246	4.7534	6.4006	4.5642	7.2635
1.1781	4.3104	5.1737	8.6607	8.5619	8.3383	5.6423	5.9330	4.6014	7.5083
4.3422	4.5302	4.8120	9.9982	5.5060	7.7867	3.8452	5.0565	3.6176	6.9885
2.2514	5.7557	1.6765	6.3451	7.1922	6.4973	5.2137	6.2869	4.5460	7.2681
2.6962	5.5282	3.3437	5.2318	8.3018	3.8564	4.6915	6.3961	4.4418	7.2709
0.8462	6.6441	1.0882	6.0259	6.0496	5.5023	5.2314	5.8576	4.7609	7.4952
-1.6368	5.7206	1.5673	7.9133	6.0561	6.6061	4.2243	5.7889	4.0785	6.1119
2.4482	2.7906	2.1654	6.4339	6.4968	6.3799	5.0703	6.1393	4.1490	7.2805

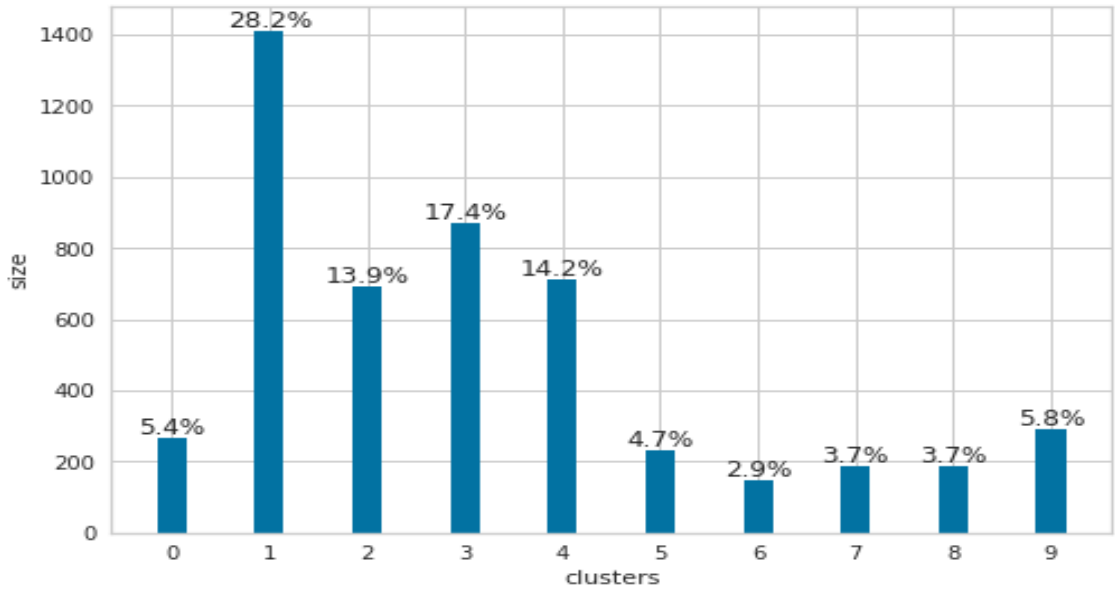
Table C.3: Random sample of UMAP-transformed event_c aggregates in 10 dimensions.

1	2	3	4	5	6	7	8	9	10
7.4735	5.4543	9.5840	5.3930	2.1095	2.3492	6.3883	4.4245	4.9597	5.5340
6.8644	3.6708	9.4579	7.1614	2.5019	2.2649	6.5195	4.4724	4.9632	5.5513
7.7347	6.9517	9.1383	5.6762	2.4310	2.3574	6.3417	4.5146	5.0155	5.5850
8.2274	5.4498	4.8121	7.4059	2.3119	2.5635	6.0882	4.2839	5.5430	5.0117
9.0415	6.0643	6.6687	4.7767	1.3913	4.0423	6.3737	3.6031	5.0327	5.6090
3.6197	5.7523	9.0883	5.5930	2.5366	2.3575	6.4873	4.7179	5.6699	5.8235
7.0208	4.9947	9.5952	6.1328	2.2983	2.3106	6.4470	4.4719	4.9709	5.5754
8.3910	8.5501	8.0182	6.1058	2.7608	2.4572	6.2376	4.5447	5.1286	5.5543
4.7262	5.4906	9.3841	6.4244	2.6524	2.3061	6.5236	4.6857	5.3878	5.8082
7.7190	5.7967	6.6610	8.0071	6.5581	1.7207	6.6749	2.9469	4.7766	5.0089
8.4711	7.2919	6.2862	8.2244	6.6728	1.8873	6.7544	3.2884	4.7098	5.1086
4.4650	6.4657	8.8304	6.0105	2.9950	2.2268	6.4012	4.6073	5.5421	5.8021
8.4089	6.1361	7.2886	7.4622	5.6082	1.6557	6.2904	3.3870	4.8029	5.1559
3.7074	5.7155	9.1581	5.6896	2.5548	2.3481	6.4961	4.7204	5.6334	5.8296
7.7302	5.8526	6.8355	8.0077	6.4000	1.7900	6.7688	2.9667	4.6810	5.0329
9.2964	5.1718	9.4853	7.1519	2.6626	2.1777	6.3847	4.4689	4.6092	5.4576
3.9403	5.3743	9.2446	6.1401	2.5972	2.3132	6.5358	4.7228	5.5504	5.8404
4.7060	7.0863	8.9875	5.9272	2.8153	2.3658	6.4474	4.7097	5.4971	5.8313
7.8098	5.9826	7.0626	7.9786	6.2277	1.8618	6.8626	2.9859	4.5749	5.0974
8.9331	4.4677	9.6562	6.4151	2.2603	2.2464	6.3928	4.3576	4.6435	5.4109
7.7563	5.9357	6.7864	8.0367	6.3986	1.8185	6.7775	3.0026	4.6881	5.0354
8.5519	5.2809	9.6935	4.2293	1.7399	2.3542	6.2952	4.3278	4.9214	5.4087
3.0943	6.0043	8.8774	5.7899	2.7339	2.3110	6.5244	4.7898	5.7642	5.9091
8.1051	2.5863	9.5603	6.8856	2.2325	2.2602	6.4827	4.2998	4.7318	5.3735
9.3345	7.8667	8.7090	5.5934	2.5573	2.2980	6.2219	4.4987	4.8990	5.4670
9.0726	5.0935	9.3750	7.7873	2.8640	2.1714	6.4203	4.5411	4.6390	5.4950
4.2730	6.6827	9.1426	6.0111	2.7842	2.3500	6.4848	4.7498	5.5280	5.8623
7.8862	6.4483	6.8282	6.7446	5.9331	1.3840	5.8040	3.3571	5.0904	5.1992
9.0544	5.1267	9.3551	7.8222	2.8880	2.1701	6.4202	4.5451	4.6452	5.4986
4.9915	6.3153	8.5713	6.2640	2.6382	2.3859	6.4339	4.6438	5.4883	5.7299
7.7970	6.4092	6.7660	6.7455	6.0023	1.3647	5.7862	3.3530	5.1242	5.2111
9.4809	7.4460	8.9196	5.6799	2.5260	2.2684	6.2391	4.4920	4.8300	5.4548
8.3103	5.5841	5.0149	7.4030	2.2935	2.5472	6.0950	4.2933	5.5108	5.0455
8.2457	4.3951	9.9850	5.2468	3.0850	1.8361	6.3515	5.1583	5.5276	5.6778
8.3505	3.1502	9.6419	6.5078	2.1499	2.2738	6.4504	4.3035	4.7026	5.3817
7.9430	5.5060	9.5904	5.4135	2.1084	2.3354	6.3699	4.4126	4.8962	5.5061

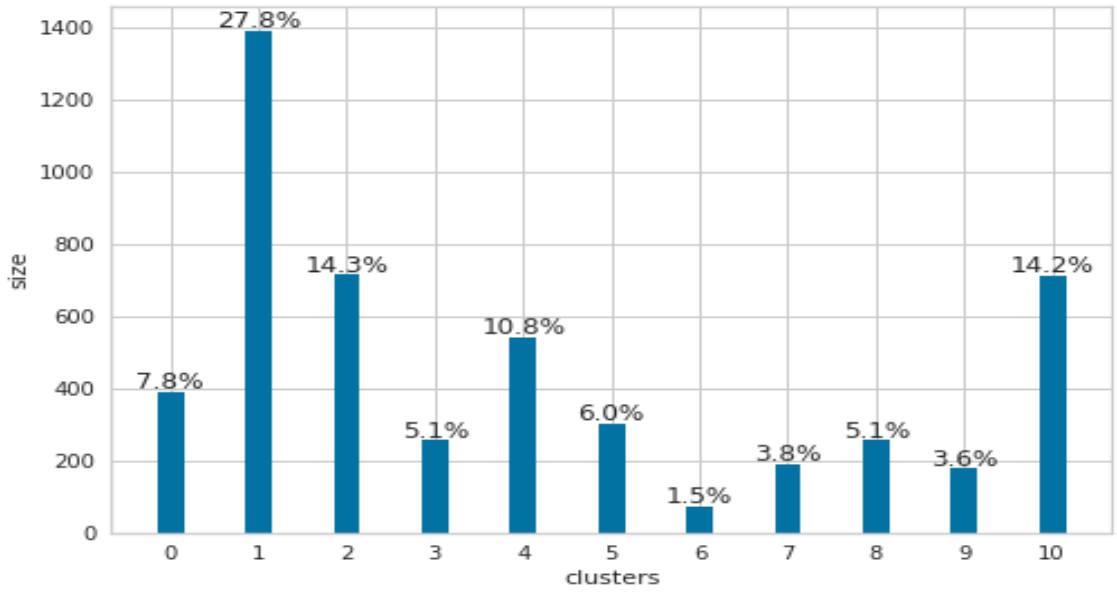
9.2768	8.4965	8.0025	6.1924	2.9247	2.3378	6.1948	4.5482	5.0034	5.4648
7.1734	6.9601	9.0875	6.8140	2.8714	2.3093	6.4222	4.6277	5.0314	5.6792
9.0710	6.1329	6.6157	4.7379	1.3782	4.0770	6.3683	3.5863	5.0403	5.6113
7.4131	4.5430	9.7761	4.8224	1.8144	2.3540	6.3827	4.3502	4.9822	5.4710
5.7540	4.8114	9.4866	6.6994	2.5351	2.2973	6.5183	4.5959	5.1604	5.7000
3.0276	6.0515	8.7173	5.6966	2.7238	2.3291	6.4992	4.7706	5.8105	5.8752
3.4353	6.0463	9.0460	5.8640	2.7106	2.3352	6.5156	4.7660	5.6816	5.8819
7.8054	2.4219	9.2328	7.6660	2.5815	2.2229	6.5295	4.3806	4.8355	5.4099
9.0027	4.3340	9.5824	7.0351	2.4426	2.2187	6.4179	4.4018	4.6140	5.4256
8.4049	7.2949	6.3013	8.2052	6.6461	1.8926	6.7278	3.3546	4.7456	5.1213
9.5144	8.2384	8.2370	6.5240	2.9335	2.3208	6.2198	4.5244	4.8867	5.4801
9.3676	8.4967	8.1594	5.8824	2.7375	2.3472	6.1934	4.4992	4.9694	5.4728
9.0444	5.0143	9.3765	7.8069	2.8624	2.1680	6.4233	4.5366	4.6415	5.4911
7.0124	4.9236	9.6103	6.0496	2.2605	2.3141	6.4447	4.4622	4.9744	5.5684

D. Clustering results: plots

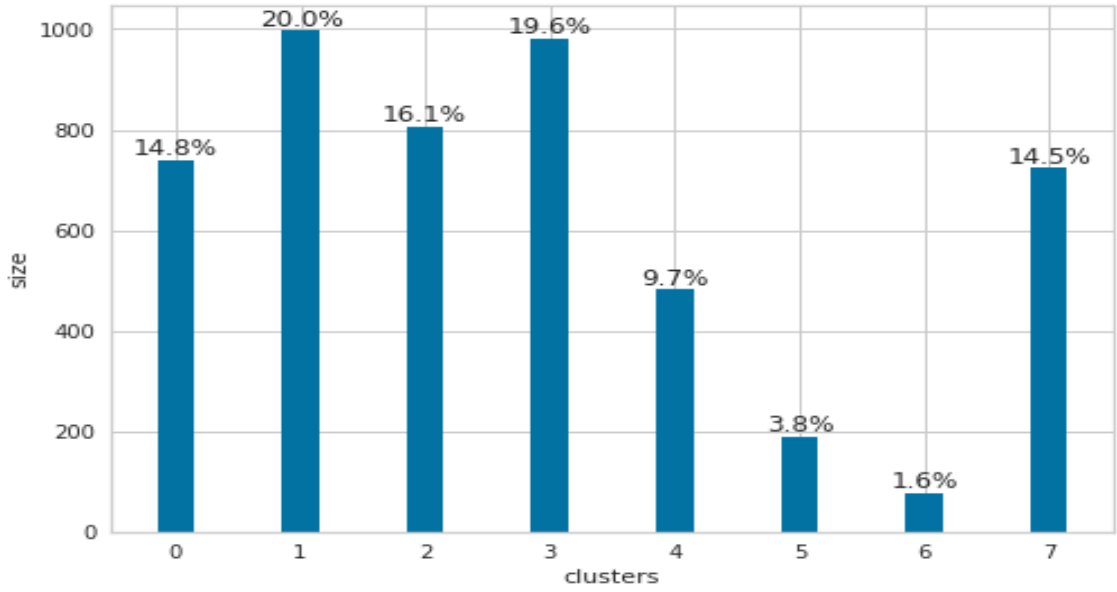
Figure D.1: Cluster sizes



(a) event_a



(b) event_b



(c) event_c