TU WIEN Informatics

Ein allgemeiner Ansatz zur Vorauswahl geeigneter Interviews bei Online-Umfragen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Paul Czapka, BSc.
Matrikelnummer 11918469

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dr.techn. Nysret Musliu
Zweitbetreuung: Johannes Klotz, Ph.D. (OGM research & communication GmbH)

Wien, 21. August 2024

Paul Czapka	Nysret Musliu

TU WIEN Informatics

A general approach to preselect useful interviews in online surveys

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Paul Czapka, BSc.
Registration Number 11918469

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Ing. Dr.techn. Nysret Musliu
Co-advisor: Johannes Klotz, Ph.D. (OGM research & communication GmbH)

Vienna, August 21, 2024

Paul Czapka	Nysret Musliu

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Paul Czapka, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 21. August 2024

Paul Czapka

Danksagung

Zu allererst will ich mich bei meinem Betreuer Associate Prof. Dipl.-Ing. Dr.techn. Nysret Musliu bedanken, der mich auf die Möglichkeit einer Masterarbeit in Kooperation mit der Firma OGM research & communication GmbH hingewiesen hat und während des Entstehungsprozesses stets für aufschlussreiche Treffen zur Verfügung stand. Meinem Zweitbetreuer Johannes Klotz Ph.D. (OGM research & communication GmbH) gilt ebenso ein besonderer Dank für die wöchentlichen Besprechungen und die daraus resultierenden hilfreichen Hinweise und Verbesserungsvorschläge. In diesem Zuge möchte ich auch der Firma OGM research & communication GmbH danken, für die Kooperation und die zur Verfügung gestellten Daten.

Dank gebührt auch meinen Eltern, die mir das Studium in Wien in dieser Form ermöglicht und mich mein ganzes Leben lang in all meinen Interessen und Aktivitäten unterstützt haben und weiterhin unterstützen.

Zu guter Letzt will ich auch noch meiner Lerngruppe, allen voran Hannes Mayrhofer und Daniela Böhm, danken, die eine große Rolle bei der Bewältigung meines Studiums gespielt haben. Ich kann mich glücklich schätzen, mit Hannes Mayrhofer einen sehr guten Freund und mit Daniela Böhm meine Freundin kennengelernt zu haben.

Acknowledgements

First, I would like to thank my advisor, Associate Prof. Dipl.-Ing. Dr.techn. Nysret Musliu, who pointed out the possibility of writing a master thesis in cooperation with the company OGM research & communication GmbH, and who was always available for insightful meetings during the development process. I would also like to extend special thanks to my co-advisor, Johannes Klotz Ph.D. (OGM research & communication GmbH), for the weekly meetings and the helpful advice and suggestions that resulted from them. In this context, I would also like to thank the company OGM research & communication GmbH for the cooperation and the data provided.

Thanks also to my parents, who made it possible for me to study in Vienna and who have supported me throughout my life in all my interests and activities.

Last but not least, I would like to thank my study group, especially Hannes Mayrhofer and Daniela Böhm, who played a significant role in helping me complete my studies. I am fortunate to have met Hannes Mayrhofer, who has become a very good friend, and Daniela Böhm, who has become my girlfriend.

Kurzfassung

Umfragen sind in verschiedenen wissenschaftlichen Disziplinen und in der Industrie von zentraler Bedeutung, da sie wertvolle Einblicke in menschliches Verhalten und Präferenzen liefern. Allerdings stehen Online-Umfragen vor wachsenden Herausforderungen durch Bots und unzuverlässige Teilnehmerinnen und Teilnehmer, die die Integrität der Daten gefährden. Wir setzen uns mit dem Problem unzuverlässiger Teilnehmerinnen und Teilnehmer auseinander, also Personen, die nicht beabsichtigen, die Umfragen ehrlich und auf Grundlage ihrer tatsächlichen Ansichten oder ihres Wissens zu beantworten. Unser Ziel ist es, eine allgemeine Strategie zu entwickeln, um nützliche Antworten im Vorfeld auszuwählen und damit die Verlässlichkeit der Umfrageergebnisse zu sichern.

Unsere Methode stützt sich auf zwei Komponenten: die Antwortzeiten auf Fragenebene und die Antwortmuster. Da beide Elemente in fast jeder Umfrage verfügbar sind, lässt sich unsere Strategie universell anwenden. Wir clustern die Antwortzeiten auf Fragenebene, um sogenannte „Speeder“ zu identifizieren, und stellen fest, dass die Betrachtung der Zeiten auf Fragenebene effektiver ist als die Analyse der gesamten Antwortzeit. Zudem nutzen wir einen Autoencoder, um Auffälligkeiten in den Antwortmustern zu erkennen, wie etwa das häufige Wählen von „keine Antwort“. Auf Basis dieser beiden Komponenten kennzeichnen wir bestimmte Interviews und bewerten unsere Strategie anhand zweier Datensätze, für die Vergleichsdaten vorliegen. Dabei übertrifft unser Ansatz traditionelle Erkennungsmethoden hinsichtlich Accuracy und Recall. Darüber hinaus ermöglicht uns eine detaillierte Analyse einzelner Interviews, das Verhalten unzuverlässiger Teilnehmerinnen und Teilnehmer besser zu verstehen. Unsere Auswertungen zeigen, dass unsere Methode in der Lage ist, nützliche Interviews vorab auszuwählen und unzuverlässige Teilnehmerinnen und Teilnehmer zu identifizieren.

Zudem untersuchen wir, welche Auswirkungen es hat, wenn unzuverlässige Teilnehmerinnen und Teilnehmer nicht ausgeschlossen werden. Dies liefert wertvolle Erkenntnisse über deren Verhalten und unterstreicht die Bedeutung dieser Aufgabe, da die Umfrageergebnisse signifikant verzerrt werden, wenn unzuverlässige Teilnehmerinnen und Teilnehmer nicht herausgefiltert werden.

Abstract

Surveys play a crucial role in various scientific disciplines and industry, offering insights into human behaviour and preferences. However, online surveys are facing increasing challenges posed by bots and bad respondents which threaten the data integrity. We aim to address the issue of bad respondents which are humans without the intention to answer the questions of a survey based on their real views or knowledge. We propose a generally usable strategy to preselect useful interviews, thus ensuring the reliability of survey results.

Our approach involves two components, the answer times on a question level and the answer patterns. Both components are available for almost every survey what makes the strategy generally usable. We cluster the answer times on a question level to detect "speeders". We find that using the times on a question level is superior to using the total response times. We try to find anomalies in the answer patterns using an autoencoder which can detect suspicious answer behaviours such as often selecting "no answer". We flag interviews based on these two components and evaluate the strategy using two datasets where a reasonable ground truth is available. We outperform traditional detection methods based on metrics such as accuracy and recall. Additionally, a case-by-case analysis allows us to gain insights how bad respondents behave. The evaluations shows that our method is capable of preselecting useful interviews and detecting bad respondents.

We analyze the effect of not excluding bad respondents. This helps to gain insights how bad respondents behave and indicates the importance of the task as there are significant shifts in the survey results if bad respondents are not excluded.

Contents

CHAPTER 1

Introduction

Surveys are one of the most common tools in scientific research and projects. Especially, in the domains health, psychology, social and market research this is a proven method to obtain scientific results and gain insights. However, there are bad respondents, as we call them, which are humans without the intention to answer the questions of a survey based on their real views or knowledge and therefore might falsify results and conclusions. In the literature, they are often called insincere, inattentive, fraudulent, insufficient effort or careless respondents, which we summarize under the term "bad respondents".

Why does the problem of bots and bad respondents appear in online surveys? There are usually two main reasons: incentives and manipulation.

Surveys often have some kind of incentive. This can be either through a monetary compensation, coupons or a lottery. This incentive might be higher than the interest of a person to accurately answer the questions of the survey. This leads to fast and basically random responding. By using bots, the profit of getting incentives can be maximized. In this case, bots will likely use some kind of random responding, too. For humans, inattentiveness or distraction can be the source of low-quality answers.

Manipulation is also a problem since some interest groups prefer specific results of a study. This can either be done by humans or bots. In this case, the answers are not random but are trying to push the result into a certain direction.

We focus on human respondents since we deal with closed survey environments where bots and manipulation are unlikely (see Section 3.2). However, bad respondents can still exist. The detection and elimination of bad respondents is a crucial task in order to keep the integrity of survey research. We propose a new method, where the response times on a question level are used for clustering to detect "speeders" and an autoencoder to detect anomalies in the answer patterns.

1.1 Goal and research questions

Every survey is different. Therefore, it is hard to provide an automated system, which detects all bad respondents across all surveys. However, the goal of this master thesis is to provide a model which excludes obvious cases and gives a preselection of suspicious interviews, which should be further checked. The approach should be applicable in closed survey environments across most surveys because it only uses the response times on a question level and the answer patterns, which are typically available.

Research questions:

- To what extent can the combination of unsupervised cluster algorithms for the response times and an autoencoder architecture for the answer patterns accurately identify bad respondents in closed online surveys?
- How does this approach compare to traditional survey quality control measures in terms of evaluation metrics?
- How big are the changes of the survey result measured with statistical significance if bad respondents are not filtered out?

1.2 Contributions

- We propose a method to preselect useful interviews by analyzing response times at the question level and examining answer patterns.
- We cluster question-level response times to identify "speeders" and demonstrate that using question-level response times is superior to using total response times.
- We use an autoencoder to detect anomalies in answer patterns, which may indicate low-quality responses.
- We test our proposed method on two different datasets, showing that it outperforms prominent existing methods in terms of recall and accuracy.
- We manually inspect interviews to gain insights into the behavior of bad respondents. For instance, bad respondents often avoid extreme answers, prefer neutral responses, and are more likely to avoid declarations. Additionally, this inspection supports our proposition that the method can effectively preselect interviews.
- We conduct a statistical analysis to determine if significant changes are observed in survey results when bad respondents are not excluded.

Parts of the work have been presented at the Austrian Statistical Days 2024 (April 4th, 2024) at TU Wien, where a vivid discussion took place, leading to numerous suggestions. Several of these suggestions have been considered and partially implemented in this thesis.

1.3 Structure of the thesis

We give a short overview of the structure and chapters of the thesis.

The thesis starts with preliminaries about unsupervised machine learning in Chapter 2. This includes an introduction regarding notation. Additionally, the algorithms which we use later in the thesis are explained.

Surveys are a complex topic. There are many things to consider and Chapter 3 tries to give an overview of how the methodology behind surveys work. We also discuss the potential challenges which can occur when conducting a survey. One challenge are bad respondents. This is the problem we try to tackle with our master thesis. There is lot of research dedicated to this topic. The field is very dynamic and there are also many recently proposed methods. Chapter 4 gives an overview of the state of the art and related work.

Chapter 5 explains the methodology of how to achieve our stated goal. The detection strategy is explained and two datasets used for evaluation as well as evaluation metrics are introduced. One dataset for testing is provided by OGM research & communication GmbH and is about the health condition of people and one dataset is a public available dataset from the literature. Our method uses two main components: the answer times on a questions level and the answer patterns. Chapter 6 is dedicated to the answer times and explains how they are used for the flagging of bad respondents. Chapter 7 discusses the answer component and how the autoencoder is used to detect anomalies in the answer patterns.

The results are shown in Chapter 8.

Chapter 9 contains a conclusion, limitations of our methodology and possible further research.

CHAPTER 2

Preliminaries: Unsupervised machine learning

Machine learning, which is a subdiscipline of artificial intelligence, has seen an unprecedented rise in popularity over the last few years due to several applications like ChatGPT. There are some names, which often come up when talking about the origins of machine learning. For example, Rosenblatt (1958), the inventor of the perceptron, is usually mentioned in this context. The development has taken place in waves because there have always been some limitations which led to so called AI winters. Fradkov (2020) gives an overview of the history of machine learning and talks about the reasons for the "gold rush of machine learning" we see today. Three main reasons are mentioned: the availability of high amount of data, the development of new algorithms, i.e. deep neural networks, and the increased computational power.

The three common sub-disciplines of machine learning are: supervised learning for labelled data, unsupervised learning for unlabelled data and reinforcement learning for environments, where an agent tries to maximize a reward. There is also semi-supervised learning, which is a mix of supervised and unsupervised learning. For our application, we deal with unlabelled data. Therefore, we focus on unsupervised methods which can be cluster algorithms or self-supervised algorithms like an autoencoder.

We use the Python library Scikit-learn by Pedregosa et al. (2011), which is a library designed for machine learning offering all different kinds of options in terms of preprocessing, algorithms and evaluation. In this chapter we present two classes of algorithms, i.e. clustering algorithms and autoencoders, which we use later on.

2.1 Notation

We introduce notation needed for the following section and used throughout the whole master thesis.

X is the data matrix. The dimensions are $n \times p$, where n is the number of observations and p is the number of features. The elements of the matrix are denoted with x_{ij}, where $i = 1, ..., n$ and $j = 1, ..., p$.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

$X_{\cdot j} \in \mathbb{R}^n$ denotes the j-th column of X for $j = 1, ..., p$.

$X_{i\cdot} \in \mathbb{R}^p$ denotes the i-th row of X for $i = 1, ..., n$.

2.2 Clustering algorithms

The goal of cluster analysis is to find groups in an unlabelled dataset. The observations of one group should be as similar as possible and the observations of different groups as dissimilar as possible. The difficulty of this task is that the number of underlying groups in the data is unknown, and it is uncertain whether any groups exist at all. The similarity between observations is typically measured by distance, e.g. the Euclidean distance.

2.2.1 k-means clustering

The book "Cluster algorithms" by Hartigan (1975) gives an overview of different distance measures and also different algorithms, e.g. k-means. The k-means algorithm is a very popular cluster algorithm. The goal is to minimize the distance within a cluster. To do this, one can take the Euclidean distances to the cluster centers, which is the mean of every observation in the cluster. The number of clusters K is a hyperparameter and has to be prespecified. We minimize:

$$\sum_{k=1}^{K} n_k \sum_{X_{i\cdot} \in C_k} ||X_{i\cdot} - \bar{x}_k||^2$$

Here, C_k denotes the set of all observations belonging to cluster k. The number of observations in cluster C_k is denoted as n_k, and $\bar{x}_k$ denotes the cluster center, which is calculated as the mean of all observations within the cluster.

$$\bar{x}_k = \frac{1}{n_k} \sum_{X_{i\cdot} \in C_k} X_{i\cdot} \qquad \bar{x}_k \in \mathbb{R}^p$$

It is not possible to solve this minimization problem with a closed form solution. Therefore, we use an iterative algorithm, which starts with random observations as cluster centers. Then all observations are assigned to the nearest cluster center and the new cluster center is calculated based on the assigned observations. This is repeated until convergence as shown in the pseudo code Algorithm 2.1 (see also e.g. Nazeer and Sebastian (2009)).

Algorithm 2.1: Pseudo code of the k-means algorithm.

Input: Data matrix X, number of clusters K

Output: Cluster assignments C_k, cluster centers $\{\bar{x}_1, ..., \bar{x}_K\}$

```
Initialize cluster centers x̄_1, ..., x̄_K (usually by randomly selecting K observations)
repeat
    foreach observation X_i. do
        Assign X_i. to the nearest cluster center x̄_k ⇒ X_i. ∈ C_k
    end
    foreach cluster k do
        Update cluster center x̄_k as the arithmetic mean of all observations in
         cluster k
    end
until convergence;
```

2.2.2 Agglomerative clustering

Agglomerative clustering is a form of hierarchical clustering. There are two options to do hierarchical clustering. One is called divisive clustering, where we start with one cluster with all observations and split this cluster into smaller clusters until the number of clusters is equal to a predefined value. The other option is called agglomerative clustering, where we start with the number of clusters equal to the number of observations n. Then, the most similar observations are combined until the predefined number of clusters is reached. Usually it is easier to combine clusters than to divide clusters. Therefore, agglomerative clustering is more efficient and more often used than divisive clustering. The merging of clusters is based on similarity. There are different options to define the similarity between two clusters. We show three of them, which all use a distance d, e.g., the Euclidean distance. Therefore, a low value means high similarity. The formulas describe the similarity between two different clusters C_k and C_l.

- Complete Linkage (see Figure 2.1): $\max_{X_{i.} \in C_k, X_{j.} \in C_l} d(X_{i.}, X_{j.})$
- Single Linkage (see Figure 2.2): $\min_{X_{i.} \in C_k, X_{j.} \in C_l} d(X_{i.}, X_{j.})$
- Average Linkage: $\frac{1}{n_k n_l} \sum_{X_{i.} \in C_k} \sum_{X_{j.} \in C_l} d(X_{i.}, X_{j.})$

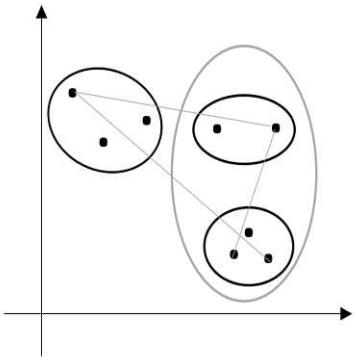

Figure 2.1: Example of Complete Linkage.

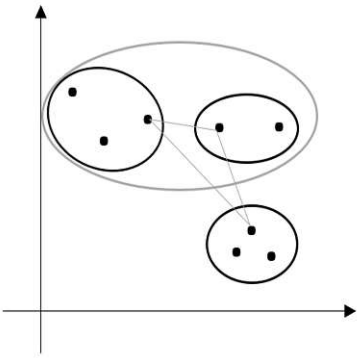

Figure 2.2: Example of Single Linkage.

When doing agglomerative clustering, the two clusters with the highest similarity based on the chosen method are combined until the predefined number of clusters is reached.

2.2.3 DBSCAN (with k-means for noise points)

Density-based spatial clustering of applications with noise (DBSCAN) was introduced by Ester et al. (1996) for spatial databases. The advantage is that the DBSCAN algorithm allows complex shapes since the algorithm relies only on the density of the points and not on a center like for example k-means. There are two hyperparameters: ϵ which defines the size of the neighborhood of each point and $Minpts$, which defines the minimum number of samples in a cluster. Two points belong to a cluster if one could reach the other point through other points which have to fulfill a density condition. We provide a formal definition.

Definition 2.2.1 (Cluster) *Given the data matrix X, the hyperparameters ϵ and Minpts, a cluster C_k is defined through the following conditions:*
1) $\forall X_{i\cdot}, X_{j\cdot} : (X_{i\cdot} \in C_k) \wedge (X_{j\cdot}$ *is density reachable from* $X_{i\cdot}$ *given* ϵ, *Minpts)* $\Rightarrow X_{j\cdot} \in C_k$
2) $\forall X_{i\cdot} \in C_k, X_{j\cdot} \in C_k : X_{i\cdot}$ *and* $X_{j\cdot}$ *are density connected given* ϵ *and Minpts*

Two points are density reachable if there is a chain of points from one point to the other where all subsequent points are belonging to the ϵ-neighborhood of the previous point (ϵ-neighborhood of p: $N_\epsilon(p) = \{q : d(p,q) < \epsilon\}$) with the additional condition that each size of ϵ-neighborhoods has to be bigger than the number of minimum samples *Minpts*. Two points are density connected if there exists another point that is density reachable from both points.

Due to the condition of the minimum number of samples *Minpts* for the ϵ-neighborhoods, every cluster has a minimum of *Minpts* points and there are also points which do not belong to any cluster. These points are called noise-points.

Definition 2.2.2 (Noise-points) *Given the data matrix X, the clusters $C_1, ..., C_k$ and the hyperparameters ϵ and Minpts, the noise-points are defined as all observations not belonging to any cluster:*
Noise-points $= \{X_{i\cdot} | \forall i = 1, ..., k : X_{i\cdot} \notin C_i\}$

We use the algorithm with survey data and have a particular interest in the noise points because these are often outliers. In survey data, too fast respondents are often outliers/noise points and therefore we extend the DBSCAN algorithm and cluster the noise points again with a k-means algorithm to distinguish between "good" and "bad" outliers because also too slow respondents might be outliers. In this way, every observation is assigned to one cluster. The number of clusters used for the additional k-means algorithm is another hyperparameter.

2.2.4 BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is an efficient clustering algorithm designed to handle large datasets by incrementally and dynamically clustering incoming multi-dimensional metric data points in a hierarchical manner. It was introduced by Zhang et al. (1996). The main advantage of BIRCH is that it can efficiently process large datasets by building a compact summary of the data using a structure called the Clustering Feature Tree (CF Tree).

Clustering Feature (CF)

The key component of the BIRCH algorithm is the Clustering Feature (CF), which is a triplet summarizing information about a cluster. For a subcluster with n data points $\{x_1, x_2, ..., x_n\}$, the CF is defined as:

$$CF = (n, LS, SS)$$

where n is the number of points in the subcluster, LS (Linear Sum) is the sum of the data points: $LS = \sum_{i=1}^{n} x_i$ and SS (Square Sum) is the sum of the squares of the data points: $SS = \sum_{i=1}^{n} x_i^2$.

The CF captures the essential statistics of a subcluster and can be used to compute the centroid, radius and diameter of the subcluster.

CF Tree

The CF Tree is a height-balanced tree with two types of nodes:

- Leaf nodes: Contain CF entries and represent subclusters.
- Non-leaf nodes: Contain CF entries that summarize information about their children.

Each node has a maximum number of entries it can hold, controlled by the branching factor B. The CF Tree is built dynamically as data points are inserted. The parameters controlling the tree growth are the branching factor B (the maximum number of children per non-leaf node) and the threshold T (the maximum diameter of subclusters stored in leaf nodes).

BIRCH Algorithm Steps

The BIRCH algorithm operates in different phases. We explain them and provide a pseudo-code in Algorithm 2.2.

Phase 1: Scanning the data and building the CF Tree Data points are inserted into the CF Tree. For each point, the algorithm traverses the tree from the root to a leaf and tries to insert the point into the closest subcluster. If the insertion causes the diameter of the subcluster to exceed the threshold T, the subcluster is split. If a node overflows, it is split, and the tree grows.

Phase 2: Condensing the CF Tree Optional phase where a smaller CF Tree is rebuilt from the original tree to remove outliers and reduce the size of the tree by increasing the threshold T.

Phase 3: Global Clustering An existing clustering algorithm, such as k-means, is applied to the leaf entries of the CF Tree to obtain the final clusters. This phase benefits from the reduced data size due to the compact CF Tree representation.

Phase 4: Refining the Clusters Optional phase where the clusters obtained from Phase 3 are refined by redistributing the data points to achieve better clustering results.

Algorithm 2.2: Pseudo code of the BIRCH algorithm.

Input: Data matrix X, branching factor B, threshold T, number of clusters K

Output: Cluster assignments C_k, cluster centers $\{\bar{x}_1, ..., \bar{x}_K\}$

```
Initialize an empty CF Tree with branching factor B and threshold T;
foreach data point X_i. ∈ X do
    Insert X_i. into the CF Tree, updating the appropriate CF entries;
end
(Optional) Condense the CF Tree by rebuilding with a higher threshold T;
Apply global clustering (e.g., k-means) to the leaf entries of the CF Tree;
(Optional) Refine the clusters by redistributing data points;
```

BIRCH is particularly well-suited for large datasets because it builds a compact summary of the data that can be efficiently clustered. The CF Tree structure allows BIRCH to handle noise and outliers effectively. The hyperparameters which must be predefined are the number of clusters, the branching factor and the threshold for splitting.

2.3 Autoencoder

Autoencoders are typically used with unsupervised data for dimension reduction, feature extraction or anomaly detection. We give a short introduction to autoencoders regarding origins and use-cases in Section 4.3.

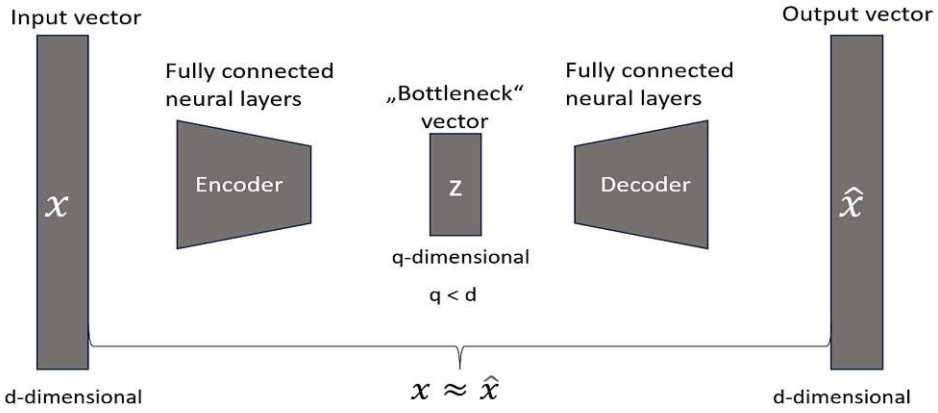

Figure 2.3: General architecture of an autoencoder.

Figure 2.3 shows the general architecture of an autoencoder, see also Bank et al. (2023) for a similar graphic and further explanations. In contrast to cluster algorithms, there

is a training process similar to supervised algorithms using the input as desired output. Therefore, it is a self-supervised method.

The input vector is p-dimensional. The dimension is decreased to a "bottleneck" vector with fully connected neural layers and then the dimension is again increased to its original size. The input and output vector should be as similar as possible. This similarity can be measured with the mean squared error (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n} ||x - \hat{x}||^2$$

This error is minimized during the training process by computing the gradient with backpropagation and is also called reconstruction error.

Autoencoders require several design choices such as the number of nodes per layer, the number of the neural layers and the choice of activation functions. Typically, the number of nodes and layers are the same for the encoder and the decoder, but they can also differ. Figure 2.4 shows how an autoencoder works with an illustration of nodes and layers that are connected with weights, which are updated in the training process. In our example, the number of layers is z and the number of nodes per layer is k_i, where $i = 1, ..., z$.

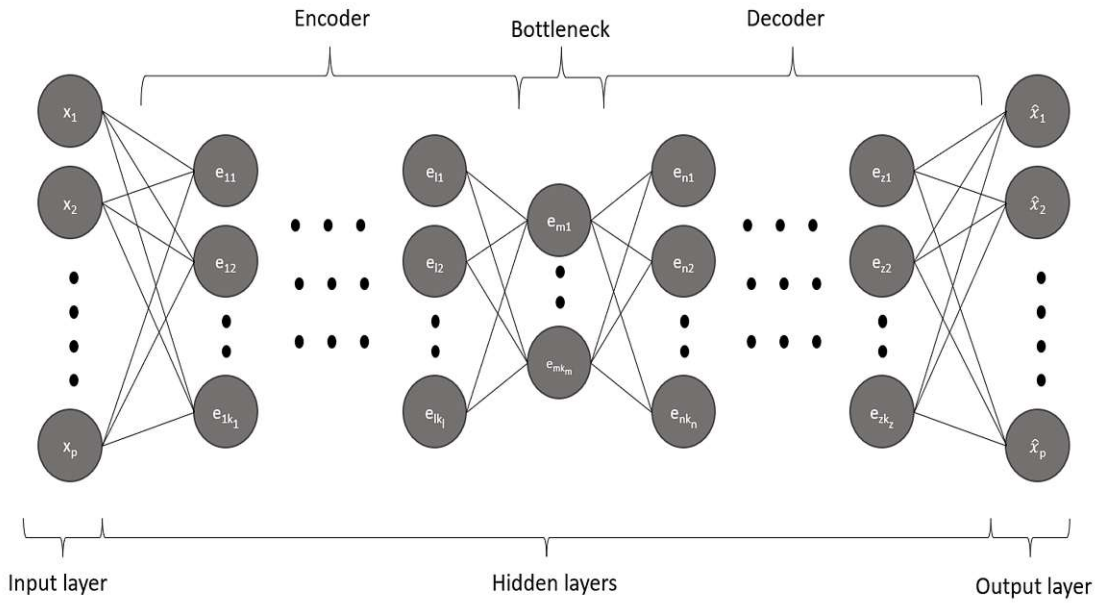

Figure 2.4: Plot of the autoencoder architecture with nodes and layers. Every line corresponds to a weight.

The first values of the hidden layers are computed with the input vector. The function g is called activation function.

$$e_{1j} = g\left(\sum_{h=1}^{p} x_h w_{1_{hj}}\right), \qquad j = 1, ..., k_1$$

After the first layer, every entry in the hidden layers is computed with all values of the previous layer and the corresponding weights.

$$e_{ij} = g\left(\sum_{h=1}^{k_{i-1}} e_{(i-1)h} w_{i_{hj}}\right), \qquad j = 1, ..., k_i, i = 2, ..., z$$

The final output is obtained with the values of the last hidden layer.

$$\hat{x}_j = g\left(\sum_{h=1}^{k_z} e_{zh} w_{z_{hj}}\right), \qquad j = 1, ..., p$$

The activation function can be different for different layers. The most common options for choosing an activation function are:

- **Sigmoid**: The sigmoid function maps the input values to a range between 0 and 1. It is defined as:
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$
- **Tanh**: The hyperbolic tangent function maps the input values to a range between -1 and 1. It is defined as:
$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$
- **ReLU**: The Rectified Linear Unit function outputs the input directly if it is positive; otherwise, it outputs zero. It is defined as:
$$\mathrm{ReLU}(x) = \max(0, x)$$

The key hyperparameters for the autoencoder are:

- **Number of Layers**: The number of encoding and decoding layers.
- **Number of Nodes per Layer**: The number of neurons in each layer.
- **Activation Function**: The function used to introduce non-linearity.

- **Learning Rate**: The step size used for gradient descent optimization.
- **Batch Size**: The number of samples per gradient update.
- **Epochs**: The number of times the entire dataset is passed through the network.

Finding appropriate values for those hyperparameters is crucial for the model's performance. Therefore, hyperparameter tuning is necessary to achieve optimal results. The choice of the hyperparameters for our use-case is explained in Section 7.2.1.

CHAPTER 3

Survey methodology overview

The aim of this chapter is to provide an overview of the broad topic of survey methodology. There are different ways a survey can be carried out, e.g., a survey mode has to be chosen. The process of conducting a survey should follow certain guidelines, which aim to ensure the integrity of the results. Even when following those guidelines, there might be challenges, which have to be tackled.

3.1 Survey modes

The survey mode describes the setting how the data for the survey is gathered. Historically, the usage of survey modes has changed. Survey research can be divided into three parts and each part is closely related to the dominant survey mode in this era like Groves (2011) explains. This mostly describes the American development. Most of the following innovations were implemented with a time delay of some years in Europe. From 1930 to 1960, mostly face-to-face interviews and mailed questionnaires were used. In the second era from 1960 to 1990, telephone interviewing was the major survey mode and also computer assisted telephone interviews became popular. Since 1990 up to now the internet has revolutionized data collection for surveys. Even though the telephone was still very popular at the start of the third era, web-based surveys are nowadays the mostly used tool. Also mixed-mode surveys, where more than one survey mode is used, are a popular option.

We provide a list with common survey modes with a short explanation, which is based on Bhattacherjee (2012). T. Smith and Kim (2015) also provide a list with advantages and disadvantages of survey modes in relation to different data collection methods (audio, visual, written). Costs are one main factor influencing the choice of a survey mode.

- Interview survey
  An interviewer, which should be trained, asks a set of predefined questions and

follows an interview protocol. If something is unclear the interviewer can clarify questions. This type of survey can be very cost-intensive.

  - Face-to-face: interview with one respondent that takes place in person
  - Telephone: interview with one respondent that is conducted by telephone

- Questionnaire survey
  A list of questions is formatted into a questionnaire, which is filled out by the respondents.

  - Mail: The questionnaire is sent to the participants via mail. After filling it out, the questionnaire is mailed back again.
  - Web-based: The questionnaire is provided online and participants answer the questions via the web.

- Mixed-mode survey
  Survey modes can be mixed in a way that a part of participants is interviewed with one mode and another part with another mode. E.g., one part receives the questionnaire via mail and another part does the web-based questionnaire. It is also possible to do sequential survey mode mixing, when one mode is followed by another after a certain time (E.g., Voorpostel et al. (2021)).

There is a discussion about whether certain survey modes are superior and if and how they should me mixed. Sakshaug et al. (2023) discuss the topic of mixing modes and biases that occur. There are also mode-effects. Greene et al. (2008) explain that telephone interviews can lead to different results when it comes to personal questions like the financial situation. When speaking with a person, people tend to gloss over certain topics. People are typically more honest in web-based surveys because they feel more anonymous.

The master thesis focusses only on web-based surveys since these surveys are the most popular nowadays and the problem we want to address is typical for online surveys.

3.2 Open vs. closed surveys

Generally, there are two modes of online surveys. The first one are open surveys, where everyone can participate. The advantage is that it is easier to get a high number of interviews. The drawback is that they are more likely to get targeted by bots or bad respondents because one can take part several times. This can lead to very poor data quality. Also, manipulation through interest groups is possible like shown in Figure 3.1. Here, an interest group instructed their members to fill out a survey based on the group's aim, which led to massive change in the result. Additionally, the control over the demographic features of the sample is very limited. Examples are entertainment polls on websites, which are not suitable for scientific research. However, unrestricted self-selected

surveys can also give insights, especially for groups, which are hard to reach. There are also intercept surveys, which pop up randomly on websites.

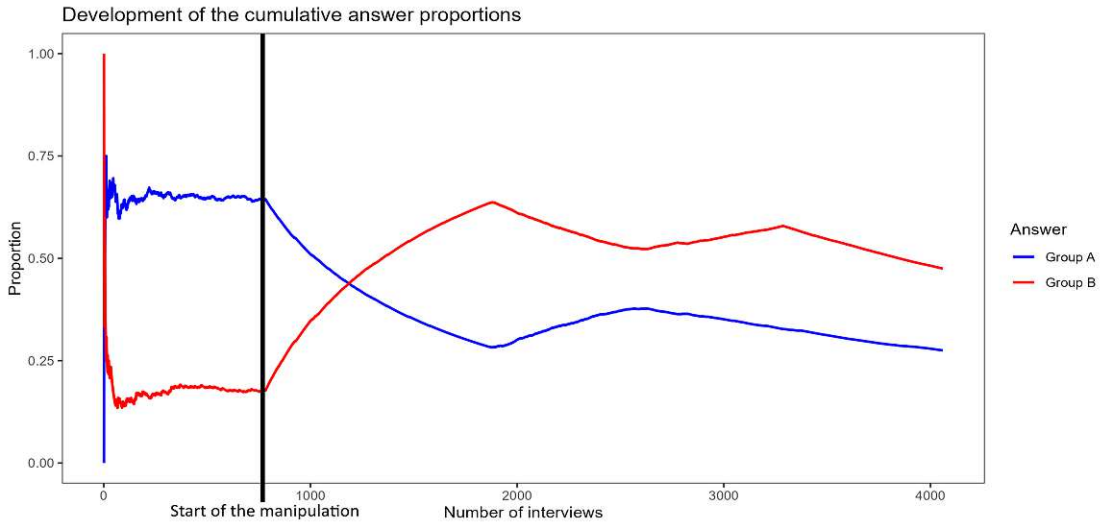

Figure 3.1: Plot of the cumulative proportions of the answers to an open survey question. The interviews are sorted by time and after around 800 interviews an interest group has tried to manipulate the result of the question. The manipulation would have been successful if not detected.

The other option is to do closed surveys where only selected and in the best case validated persons can participate. If possible, this is the better option, but you need a sampling frame where you know the contact details, which is often not available. There are different options to do closed surveys. An example is list-based sampling with a list of e-mail addresses or with a panel. A panel can be pre-recruited, where selected people are asked if they want to participate in a series of online surveys. The selection of those people should be based on probability based sampling. It is also possible that a panel is following the opt-in approach, where volunteers choose to participate. The company OGM research & communication GmbH uses the approach of closed surveys with an online-panel where people have to register and are checked. Registration is only possible after participating in an open survey or after getting recruited through a telephone call.

This summary of possible online surveys is based on Fricker Jr (2016), where also more details about the described methods including examples can be found.

3.3 Conducting a survey

The process of conducting a survey should follow a certain pattern. This is shown in Figure 3.2, which is a slight adaption of the steps described by Kelley (2003) for scientific research. Gaur et al. (2020) also provide similar guidelines. Basically, the process should

be the same if a survey is done for scientific research or done for other reasons, e.g., by a company for market research.

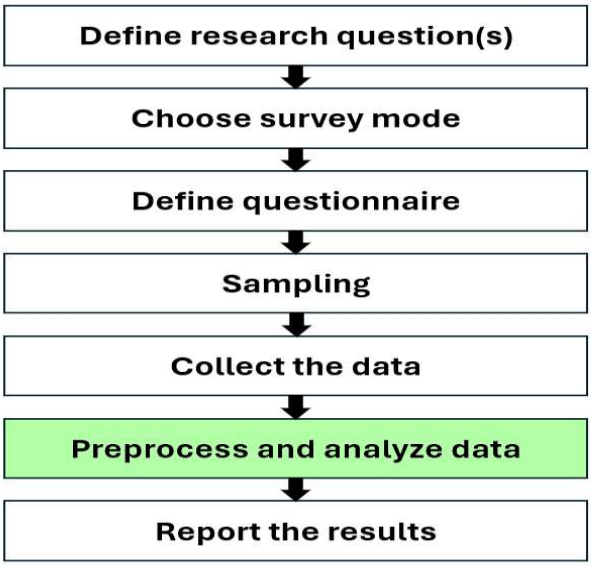

Figure 3.2: This plot shows an overview of the steps to conduct a survey. The step "Preprocess and analyze the data" is highlighted because this is the part where the master thesis is particularly relevant.

We give a short explanation for each step. There are also guidelines designed through a cooperation of the ESOMAR - the European Society for Opinion and Market Research and the WAPOR - the World Association for Public Opinion Research (2014), which should be followed.

3.3.1 Define research goals

It is important to have a clear goal in terms of which questions the survey should be capable of answering. Therefore, the research questions should be defined as a first step. They should be clear and it should be possible to use the methodology of a survey to answer the questions. This applies for scientific research as well as for all other areas where surveys are used.

3.3.2 Choose survey mode

In Section 3.1 the possible survey modes are described. Every survey mode has advantages and disadvantages. The optimal survey mode might vary across different surveys. One should know all possibilities to opt for one mode. For web-based surveys, additional options are described in Section 3.2. The possible survey modes might also depend on the availability of contact details in the sampling frame.

3.3.3 Define questionnaire

The quality of the results highly depends on the step of defining a questionnaire. The survey questions should be asked in a way that the results can be used to answer the predefined research questions from the first step. Roopa and Menta Satya (2012) describe the steps to design a questionnaire. Designing a questionnaire goes hand in hand with a high number of possible options. Which questions should be asked? How many questions should be asked? How should the questions be formulated? And several other considerations have to be kept in mind. The language should be as simple as possible. Questions should discriminate between answer groups in order to be able to make statements.
Often, a pretest is done to get an idea, which questions should be used.

There are different types of questions. The two most important ones are:

- Closed-ended questions: The answers are predefined. Closed-ended questions are for example yes/no questions, questions on a Likert-scale or multiple choice questions.
- Open-ended questions: The questions should be answered in free form.

Regarding the topic of the master thesis, it is also possible to add questions that aim to find inattentive or bad respondents. These can for example be a questions, which tell you which answer you should choose. These questions can be used as an indicator for filtering out respondents. Also questions with logical dependencies can be used, e.g., reverse worded items, where the same question appears twice but with a reversed formulation.

3.3.4 Sampling

Särndal et al. (2003) define the goal of a survey as making statements or providing information about a finite population or a subpopulation of special interest. However, a survey is typically not conducted with a whole population. Therefore, a sample has to be drawn from the population, which allows conclusions about the whole population up to a statistical error. This is usually done with probability sampling or quota sampling. Every person of the available population gets a non-zero probability of getting drawn into the sample. Demographic features and response rates influence this probability for every person. Särndal et al. (2003) provide a lot of statistical details about the sampling methodology. The ultimate goal is that the distribution of the final sample is as similar as possible to the whole population.

3.3.5 Collect the data

Depending on the survey mode, the sample has to be contacted and the contacted persons should provide answers to the questions. One must keep track on the number of contacted people and the response rates.

3.3.6 Preprocess and analyze data

After the data collection is done, the data has to be preprocessed and analyzed. The preprocessing step is where this master thesis comes into play. Based on the quality of the interviews, we want to filter out certain respondents. There are proposed methods to do this like mentioned in Section 4 and our approach is described later. Preprocessing can also be the inspection and cleaning of open-ended answers.

Depending on the survey, missing values have to be handled in the preprocessing step. The respondents with missing values can be excluded or imputation of the missing values has to be done. Imputation strategies can be random but usually use additional information to get a prediction for the missing value.

Weighting of respondents based on demographic features to achieve the proportions of the original population is often done. Then, the percentages of the answers for each question can be calculated for closed-ended questions. Open-ended questions are mostly manually inspected and similar answers can be categorized. In general, it depends on the whole survey design, how the data is analyzed. This might vary based on the initial goals. Statistical testing is also an option.

3.3.7 Report the results

The report of the results depends on the context. For scientific research, the whole survey process has to be reported in detail. If a survey is done for a company, the results with additional information but less details might be enough. Usually, this is defined beforehand with the client. Overall, the process should be transparent. This topic is also addressed in the guidelines mentioned at the beginning of Section 3.3.

3.4 Challenges for online surveys

Nowadays, online surveys are the most used survey mode due to several reasons, e.g., reduced costs and easier analysis. Also, the number of people using the internet has increased in the last two decades. Figure 3.3 shows the development of households with internet access in Austria in the years from 2002 to 2023. The percentage increased steadily and 2023 95% of the households surveyed had access to the internet. But online surveys also face some challenges. Some of the challenges occur in every survey mode like non-response bias. However, there are also specific challenges for every survey mode. We describe some common general and specific challenges for online surveys.

3.4.1 Response rates

The response rate is the proportion of selected people, who take part in the survey. The response rates and the factors influencing the response rates are subject of many studies. From our experience at the company OGM research and communication GmbH, response rates vary especially across age groups, which has to be accounted for in the sampling

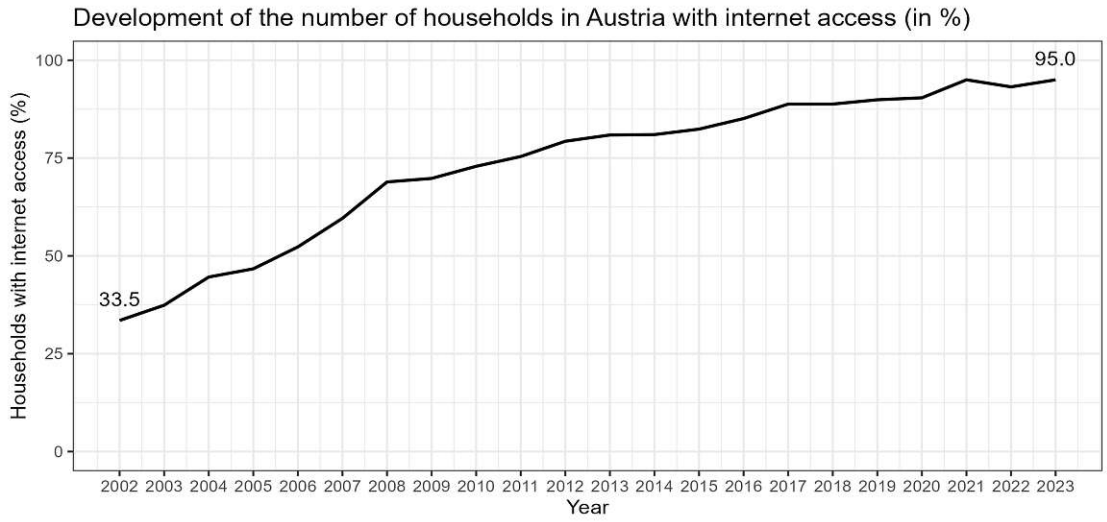

Figure 3.3: Plot of the development of the access to internet in Austrian households in the age group ranging from 16 to 74. The data is from the survey on the use of Information and Communication Technologies (ICT) in households by the Austrian statistic institute Statistik Austria (2024).

process to obtain a representative final sample.
When online surveys were first introduced, the response rates were found to be lower than for other survey modes (Manfreda et al. (2008)). More recent studies confirm that this is still true. Daikeler et al. (2020) find in a study that the response rates are on average 12% lower than for other modes for 114 examined studies. A common way to improve response rates are incentives, which can be monetary, coupons or some sort of a lottery. The design of incentives can be challenging in order not to attract specific groups. Lipps et al. (2019) discuss the issues and give a guide to answer the questions if and how incentives should be used.
One might think just inviting more participants accounts for this issue. But there is another problem associated with low response rates, namely non-response bias, which leads us to the next section.

3.4.2 Bias

A bias occurs when the sample is not representative of the population. Fricker Jr (2016) describes issues that can appear in online surveys including biases. We present two of them.

Non-response bias

It is possible that one group in the population is less likely to take part in the survey, i.e. the response rate for this group is lower. If this is the case and it is ignored, then this

introduces a non-response bias. There are several ways to address this issue. Särndal et al. (2003) also talk about this topic regarding the sampling process. For panel based surveys, one should keep track of the response rates in the demographic groups to be able to address this issue already in the sampling process. Otherwise, this group is underrepresented in the final sample. If the group responding less likely is known, e.g., a certain age group, the non-response bias can be reduced or eliminated with weighting. However, it is also possible that a feature which is not included in the data like ideology leads to a lower response likelihood. In this case, the non-response bias is difficult to address.

It is worth noting that also difficultly formulated questions or questions which are too sensitive can lead to people not answering a question. Therefore, the design of the questionnaire has an impact on the non-response. This can also lead to a measurement error if people do not understand a certain question or refuse to answer accurately.

A similar bias, which is closely related to the non-response bias, is the bias that occurs when there is a predominant group dropped through manipulation checks. Varaine (2023) explains this. The effect of underrepresentation of this group is the same, which can lead to false conclusions. We have to keep this in mind throughout the master thesis and make sure not to drop too many subjects and also to examine the dropped respondents further to find potential patterns.

Selection bias

Selection bias is one main argument against online-only surveys and why mixed-mode surveys are used, where for example a part of the survey is conducted with a telephone survey and another part with a web-based survey. A selection bias occurs, when there is one group, which is not able to take part in the survey. For online surveys, these are typically people without access to the internet or people not using the internet. Again, this group is missing in the final sample and the results might be biased. As mentioned before and shown in Figure 3.3, the size of this group gradually decreased over the last years, but the group is still present.

3.4.3 Bots

Bots are an issue limited to online surveys. However, the occurance highly depends on the way the online survey is conducted. Section 3.2 describes the options and especially open surveys can be easily targeted by bots. The number of bots has increased a lot in the recent years. Pinzón et al. (2023) analyze 36 published studies and the average number of valid responses has gone down from around 75% before 2019 to around 30% in 2022. There are strategies to collect data in a way that bots are less likely. Verification of personal information like Glazer et al. (2021), comparing answers to administrative data like Alvarez and Li (2023) or distributing the survey with a list of verified e-mail addresses like Wardropper et al. (2021) reduce the risk of bots. CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart) is widely used and

protect against some bots but modern bots are able to circumvent the tests (Goodrich et al. (2023) and Griffin et al. (2022)).

Strategies to detect bots are described in Section 4.1.

3.4.4 Bad respondents

Bad respondents are the challenge we want to address with this master thesis. They are also sometimes called inattentive, careless or insincere respondents. As already explained in the introduction, they do not have the intention to answer the questions of a survey based on their real views or knowledge and therefore might falsify results and conclusions. Existing strategies to detect those respondents can be found in Section 4.2.

CHAPTER 4

State of the art and related work

It is desirable to filter out human respondents, who are inattentive or have some kind of bad intention in mind, and bots. The behaviour of these types can be similar but might vary a lot. We give an overview of existing methods in the literature.

4.1 Bots

Bots are not the main focus of this master thesis, but some bot detection strategies can be used for detecting bad respondents and vice versa. Currently, there is a lot of research going on to detect bots since they are a growing problem. Nowadays, the quality of bots is improving very fast and more sophisticated bots are hard to detect. "As we learn about bots, they learn about us." (Storozuk et al. (2020)). Bonett et al. (2024) compare an own detection strategy, which involves the verification of answers by human beings, with the fraud detection system used by the widely used survey platform Qualtrics, which incorporates a wide range of bot detection methods. Both systems have rather low agreement on whether an interview is a bot or not, indicating the difficulty of the task.

Researchers are facing this problem and try to raise awareness on this topic. To detect bots paradata like IP-address, geolocation or other respondent statistics should be combined alongside other methods like checking for inconsistency in the answers (Goodrich et al. (2023)). Often, open-ended questions with the exact same wording as response over many interviews are the reason why researchers get suspicious and investigate further like done by Griffin et al. (2022). Pinzón et al. (2023) is a recent publication, which gives an overview, explains and tests the most common techniques to detect bots. They find that certain patterns in e-mail addresses are often a good indicator to detect bots. Additionally, they and the literature in general suggest using a combination of all available different features.

4.2 Bad respondents

Our focus is on humans. Ternovski et al. (2022) find that the number of bad respondents increased since 2020 indicating the necessity to filter them out in order to avoid biases. Typically, the response times and the answer patterns are used to find human respondents which should be excluded. Curran (2016) summarizes different methods like filtering out based on the response time or outlier analysis using the Mahalanobis distance. Also, other methods like a measure for the individual answer consistency, long-string analysis, or a question to self-report the attentiveness or reverse worded items are used. They recommend a multiple hurdles approach. This means that not only one indicator, but a combination of indicators should be used. A lot of these measures are implemented in the R package "careless" by Yentes and Wilhelm (2023). We explain the most common methods. In Chapter 8, some of these methods are used for comparison to our own strategy.

- **Long-string analysis** (Johnson (2005)) checks how often the same answer is chosen consecutively. Longer strings indicate potential inattentiveness or repetitive responding.
- **Intra-individual response variability (IRV)** (Dunn et al. (2018)) analyzes the variability in an individual's responses. Low variability suggests insufficient effort or inattentive responding.
- **Psychometric synonym/antonym** (Meade and Craig (2012)) examines the consistency of responses to synonymous or antonymous items. The correlation between two questions is calculated to find questions with high or low correlations indicating a dependeny between these two questions. If the answers of a respondents do not follow those correlations, this may indicate careless answering.
- **Mahalanobis distance** (Mahalanobis (1936)) is used to identify outliers. Meade and Craig (2012) use this statistical distance to detect multivariate outliers, which may indicate abnormal or inattentive responses.
- **Even-odd consistency** (Johnson (2005)) assesses the consistency between responses to even-numbered and odd-numbered items. High inconsistency can signal careless responding.
- **Z_h statistic** (Drasgow et al. (1985), Felt et al. (2017)) evaluates person-fit by comparing an individual's response pattern to the expected pattern based on item parameters.
- **Total time** is used to filter respondents based on their total response time. This helps to identify those who may not be providing thoughtful answers. A threshold for exclusion has to be chosen. E.g., 0.4 times the median total response time (Greszki et al. (2015)) is an option. It is also possible to use more complex thresholds (Soland et al. (2021)), e.g. for filter questions.

The popularity of machine learning has also made its way into survey research. Buskirk et al. (2018) give a general introduction to machine learning methods for survey researchers. Alongside other applications, machine learning is also used to detect bad respondents. Jebreel et al. (2020) only consider the answers and use the unsupervised cluster algorithms DBSCAN, PCA and isolation forest. They evaluate the method on a dataset where they simulate responses to get bad respondents. However, Schroeders et al. (2022) find that simulation studies differ from real world settings as the human behaviour is hard to simulate. They use gradient-boosted trees as well as traditional methods to identify careless respondents and also provide a dataset with labels. Read et al. (2021) focus on question-level response times, using them to perform PCA for dimensionality reduction, followed by clustering with expectation maximization (EM-clustering).

A model based approach to recognize patterns in responses like Ulitzsch et al. (2022) or with the incorporation of response times like Ulitzsch et al. (2024) can also be used. However, they state that the models typically do not generalize well across surveys because of the difference in surveys, which makes this approach less attractive.

Most of these methods only work for closed survey questions, where there are fixed answers and no answers in natural language are required. The master thesis also does not include the analysis of questions with natural language as answers. However, there is the possibility to use these questions in order to detect bad respondents. Kennedy et al. (2021) focus on open-ended questions and incorporate additional paradata like IP-addresses to detect suspicious interviews.

Varaine (2023) highlights that dropping too many interviews might also be a problem because this can lead to a bias. This problem is already mentioned and explained in Section 3.4.2.

4.3 Anomaly detection with autoencoders

We want to use an autoencoder architecture to find anomalies in the answer patterns. The concept was first introduced by Rumelhart and McClelland (1987) and gained popularity in the last years especially in the field of image processing. Chen and Guo (2023) review recent developments regarding the use of autoencoders. Kieu et al. (2022) try to make autoencoders for time series outlier detection more robust and explainable. One application is also anomaly detection or fraud detection. E.g., Gomes et al. (2021) use the reconstruction error of an autoencoder to detect insurance fraud.

Anomaly detection is also the use case for our task. We use an autoencoder to detect anomalies in the answer patterns. Random responding or choosing always the first or last answer might lead to an anomaly. However, it is crucial to keep in mind that an anomaly per se is not a bad thing because it might be possible that one person just gives completely different answers than the rest, what might also lead to an anomaly. E.g., this can happen if the questionnaire is about health conditions and there is a respondent in very poor health.

The following chapters explain our strategy to detect and preselect potentially bad respondents based on the response times and the response patterns. The response times and answer patterns are available for almost every survey, which supports the goal to provide a generally usable approach.

CHAPTER 5

Methodology

This chapter describes the methodology we use to try to achieve our stated goal. We want to provide a generally usable approach to preselect useful interviews. Our general approach is based on two main components, which are available for most surveys, the answer times on a question level and the response patterns. Figure 5.1 gives an overview of the strategy. We cluster the response times on a question level to detect "speeders" and use an autoencoder to detect anomalies in the answer patterns. These two components are then again combined and give a validity score which can be used for a preselection of useful interviews.

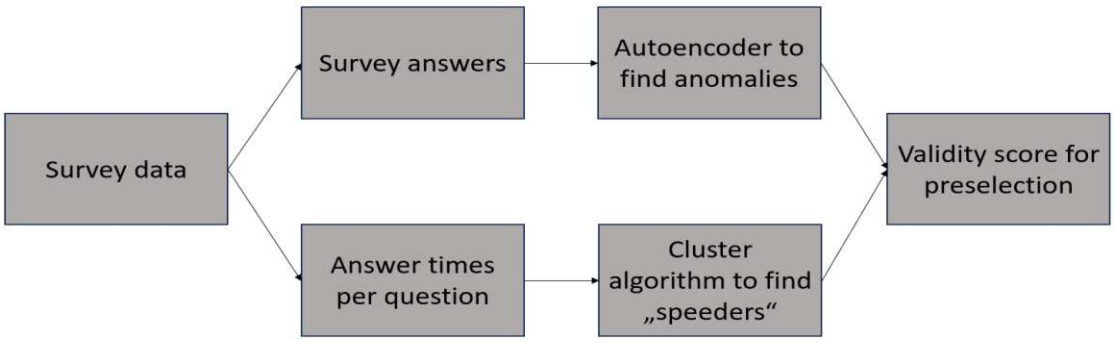

Figure 5.1: Overview of the methodology.

5.1 Datasets

We need data to implement and test the methodology. The master thesis is written in cooperation with the Austrian company OGM research & communication GmbH, which is an institute for opinion research doing surveys on a regular basis. Therefore, we

have access to surveys, which can be used for the master thesis. Additionally, there is a dataset available from Schroeders et al. (2022), where some participants of a survey were explicitly told to answer the questions as if they would not care about the survey.

5.1.1 Dataset 1: Survey for the Gesundheit Österreich GmbH

This is the main dataset of the master thesis. The dataset is provided by the company OGM research & communication GmbH, who did the survey on behalf of the Gesundheit Österreich GmbH. All plots shown in Chapter 6 and Chapter 7 are generated with this dataset. The survey is in German and asks some general questions about demographic details like age or education, but most questions focus on the health condition of people. It was conducted in 2023 and selected people were asked online or via mail to answer the questions. We only include the online interviews in our analysis, because for the offline interviews there are no response times, which are used for the detection strategy. Around two thirds of the online questionnaires were filled out with a smartphone. Table 5.1 summarizes the main features of the dataset.

Attribute	**Value**
Number of interviews	1028
Labels for exclusion based on other checks	22
Number of total questions used	31
Number of open questions (numeric)	2
Number of single choice questions	24
Number of multiple choice questions	5

Table 5.1: Details about Dataset 1.

Like for basically every survey, there are no true labels for our task available. We still need some kind of ground truth to test and evaluate our method. Therefore, we rely on other checks based on the expertise of OGM research & communication GmbH, which give us labels for inclusion or exclusion of an interview. 22 out of 1028 interviews have a label for exclusion based on other checks. The checks include a test to find "speeders" and a test to exclude respondents who choose the answer "no answer" very often.

5.1.2 Dataset 2: Literature survey

The second dataset we use is a dataset provided by Schroeders et al. (2022). They did a survey where some participants were explicitly told to answer the questions carelessly. This leads to a dataset with labels for every respondent. It can be criticised, and this is also mentioned by the authors, that one cannot completely rely on the labels because people might have ignored the instruction. However, this is still a good benchmark, where different methods can be compared. Unfortunately, the answer times are not available on a question level, but on a page level and there are only six pages. Therefore, we can only

use the response times on a page level for clustering. Table 5.2 summarizes the main features of the dataset.

Attribute	Value
Number of interviews	605
Labels for exclusion	244
Number of total questions used	60
Number of open questions (numeric)	0
Number of single choice questions	60
Number of multiple choice questions	0

Table 5.2: Details about Dataset 2.

The distribution of attentive and inattentive respondents is not realistic. For this reason, we create bootstrap samples with a more realistic ratio of interviews that should be included and excluded.

5.2 Examine answer times with cluster algorithms

We use the answer times on a question level. Typically, some preprocessing like data imputation and log-transformation is needed. Then, we use a similar approach like Read et al. (2021). They perform PCA and do EM-clustering. We perform PCA and apply different cluster algorithms with different parameter settings. Based on the silhouette score we decide which cluster algorithm to use. With a threshold, we get some clusters, where the participants of the cluster are considered as "speeders" and get a flag for the time component. This approach is described in detail in Chapter 6.

5.3 Examine answer patterns with an autoencoder

We use the answers of one interview as input vector. We preprocess the data. Then, the encoder reduces the dimension of the vector to a bottleneck and again increases the dimension of the vector with a decoder to its original size. The new vector can be compared to the original vector and should be as similar as possible. The difference between the original and reconstructed vector can be measured with the mean squared error of all components and is called reconstruction error. This reconstruction error is minimized in the training process. With cross validation we ensure that each interview is not involved in the training process to get its own reconstruction error. Interviews with a high reconstruction error do not follow the majority of the data and have some kind of anomaly. We again define a threshold to flag observations with the highest reconstruction errors for the pattern component. Our approach is similar to Gomes et al. (2021). The details are described in Chapter 7.

5.4 Combining times and patterns

Afterwards, we combine the two indicators. When an interview is flagged by both indicators, one can quite confidently eliminate this interview. When only one indicator is suspicious, further investigation is needed. This further investigation can be either done manually or with other available checks. The three possible outcomes are:

- No flag for either component → no further checks
- Flag for only one component (answer times/answer patterns) → further inspection/checks
- Flag for both components → exclusion of the interview

5.5 Evaluation

The task is unsupervised and we only rely on unsupervised methods. Since we have labels for our two datasets and since this is a classification problem, evaluation metrics like accuracy, recall, precision and F1-score can be calculated for different methods.

We use a confusion matrix shown in Table 5.3, which provides a detailed breakdown of the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These elements are defined as follows:

- **True Positives (TP):** The number of correctly identified implausible cases.
- **True Negatives (TN):** The number of correctly identified plausible cases.
- **False Positives (FP):** The number of plausible cases incorrectly identified as implausible.
- **False Negatives (FN):** The number of implausible cases incorrectly identified as plausible.

	Predicted Positive	Predicted Negative
Actual Positive	TP (True Positives)	FN (False Negatives)
Actual Negative	FP (False Positives)	TN (True Negatives)

Table 5.3: Structure of a confusion matrix.

We use the confusion matrix to calculate the following metrics:

- **Precision:** The ratio of correctly predicted positive observations to the total predicted positives. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** Also known as sensitivity, it is the ratio of correctly predicted positive observations to all observations in the actual class. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **Accuracy:** The ratio of correctly predicted observations to the total observations. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **F1 Score:** The weighted average of precision and recall. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Our method will be tested and evaluated on the two described datasets in Section 5.1. For this task the recall (sensitivity) is most important since we do not want to miss any bad respondents (Pinzón et al. (2023)). In Chapter 8, we also try methods to detect implausible interviews known from the literature and compare the results to our findings.

In addition to the quantitive evaluation, a qualitative evaluation by manually inspecting misclassified observations is done.

Furthermore, statistical tests will be conducted to check the impact of excluding respondents on the aggregate survey results. This is done with bootstrapping. Bootstrapping can be used for a variety of different use cases. Hastie et al. (2009) give an introduction how bootstrapping works in general. We use it like this: we generate bootstrap samples with all interviews and bootstrap samples with only the "good" observations. Then, the bootstrap samples are evaluated and we examine if there is a difference in the result distribution. This is done and further explained in Section 8.3.

CHAPTER 6

Response times

The response times are typically saved for online surveys and often used for quality checks. Some common checks only include the total response time. We use the response times on a question level, which give a much better picture of the response behaviour. In this chapter we explain how the response data is distributed, which preprocessing steps we take and how the data is clustered to detect potential low quality respondents.

6.1 Distribution and analysis of response times

To get an idea how response time data look like, we provide plots to examine typical distributions. We do this with the dataset provided by OGM research & communication GmbH, which is described in Section 5.1.1.

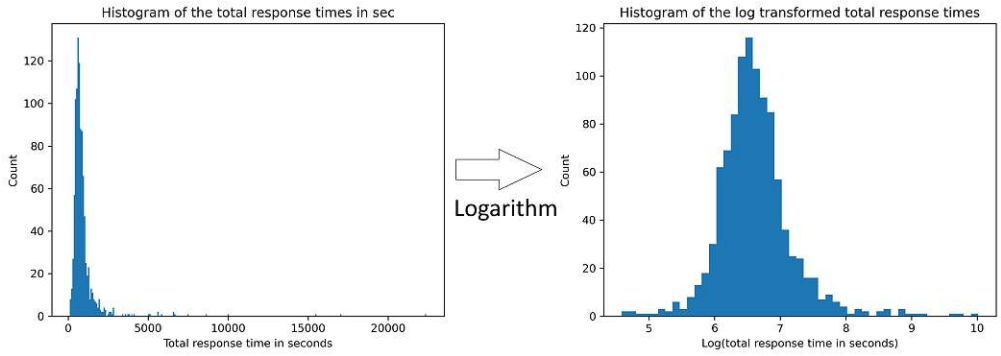

Figure 6.1: The left panel shows a histogram of the total response times in seconds. The right panel shows a histogram of the log-transformed total response times.

Figure 6.1 on the left shows that the total response times have a right skewed distribution. This is due to the fact that some people take very long for the completion of a survey because they might get distracted and do something else in the meantime like answering a phone call before finishing the survey. With a log-transformation we obtain a distribution closer to a normal distribution which is shown on the right of Figure 6.1.

We use the response times on a question level and therefore also inspect the distribution on a question level. The distribution is similar to the distribution of the total response times even though we do not achieve normality as well as for the total response times with a log-transformation. Figure 6.2 shows the response times for Question 2, which is an open question and asks about the age of the respondents in years.

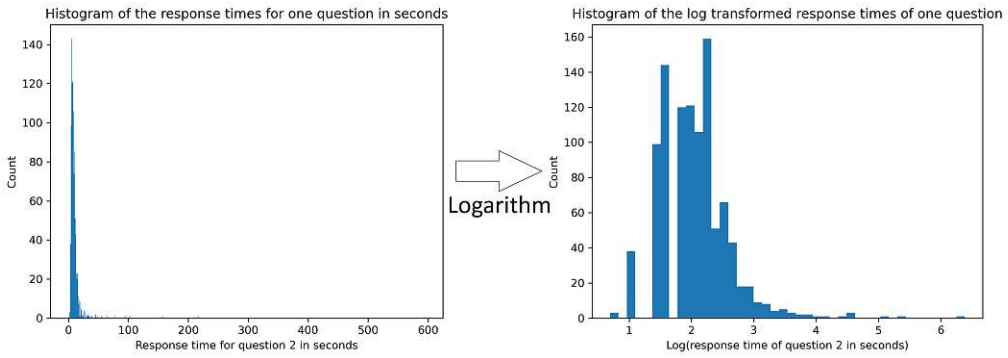

Figure 6.2: The left panel shows a histogram of the response times of Question 2 in seconds. The right panel shows a histogram of the log-transformed response times of Question 2.

The further preprocessing steps will be explained in Section 6.2.

We analyze factors which might influence the response times. Even though we do not use this information in our application due to the reason that it assumes the availability of additional information, the findings can be used if the corresponding data is available. E.g., the clustering could be done for different age groups independently. Table 6.1 shows that this might indeed be useful since the oldest group takes around 47% longer than the youngest group to complete our example survey.

Age group (years)	n	Median total interview time (seconds)
16-29	89	549
30-39	126	612
40-49	143	604
50-59	211	713
60+	457	805

Table 6.1: Median time by age group.

If we do not only use the median but the whole distribution of the total response times, the message that young people are faster stays the same as shown in Figure 6.3. Not only the median is lower for the younger group, but also the proportion of very fast respondents is higher.

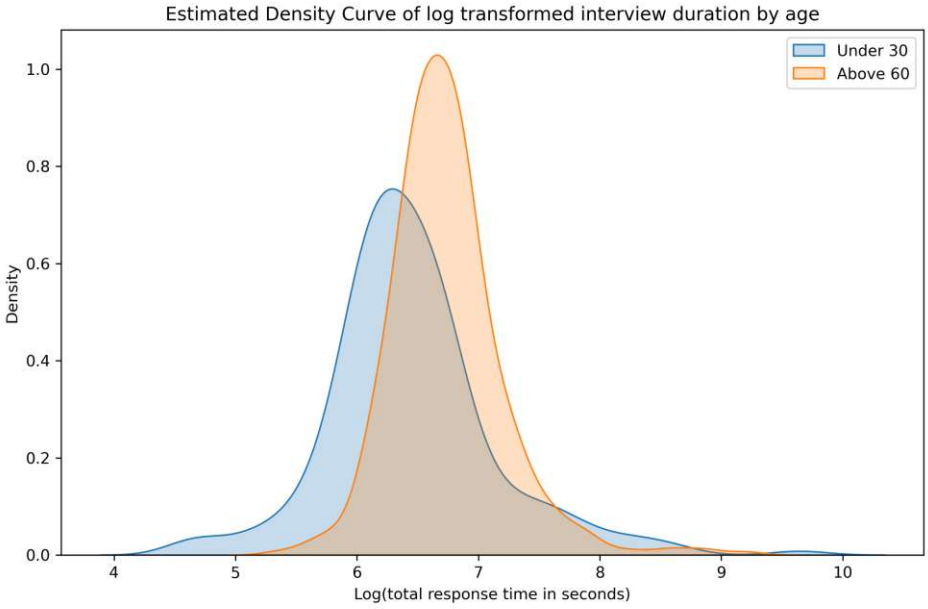

Figure 6.3: Density estimation for the youngest group (under 30) and for the oldest group (above 60) for the total response times.

Young people under 30 have grown up with digital technology (digital natives) and are more comfortable and efficient in navigating online interfaces, especially when using a smartphone. Additionally, cognitive processing speed tends to decline with age, so younger people can process information and make decisions more quickly.

Also, the device influences the total response time. This is shown in Table 6.2. Not surprisingly, people using mobile devices like smartphones are the fastest.

Device	n	Median total interview time (seconds)
Desktop	450	751
Mobile	562	696
Tablet	16	724

Table 6.2: Median time by device.

6.2 Preprocessing

Before we can apply clustering algorithms, we have to perform some preprocessing steps.

6.2.1 Imputation of missing values

There is the possibility that a question is not shown to every participant. These questions are called "Filter questions" and are only shown based on a certain condition like the answer to another question. This can lead to missing values. We still want to use the information of this column. Therefore, we impute the missing values. There are different strategies to do this. Jadhav et al. (2019) compare the performance of different strategies. It is possible to do single imputation where every missing value of one variable is imputed with the same value. This is typically the mean or the median. Then, there is multiple imputation where the imputation of every missing value can differ. This is often done with some kind of prediction model.

We use multiple imputation with the median of a variable and correct it for each participant with the median deviation of this person in other questions. This is a robust approach and yields plausible imputations. We denote the matrix with the response times as $T \in \mathbb{R}^{n \times p}$, where every column is a question and every row is a participant. The imputation includes the assumption that some people are generally faster and some are slower in responding. The median deviation should account for this. Equation 6.1 shows the strategy we use.

$$\forall i, j : T_{ij} \text{ is a missing value} \Rightarrow$$

$$\hat{T_{ij}} = med(T_{.j}) + med(\{T_{i1} - med(T_{.1}), T_{i2} - med(T_{.2}), ..., T_{in} - med(T_{.n})\}) \tag{6.1}$$

This approach can be viewed as analogous to a random effects model, where individual deviations of the response speed are treated as a random effects, capturing the variability across participants.

6.2.2 Data transformations

In Section 6.1 we already saw that a log-transformation is useful. Therefore, we apply a log-transformation for every question like shown in Figure 6.2. Even after the log-transformation, there are often outliers on the right side. These outliers might have a negative impact on the clustering algorithms. For this reason, we come up with a strategy, which reduces the effect on the upper outliers and shifts the focus on the fast respondents. This can be achieved with a reciprocal transformation: $T_{ij}^{new} = \frac{1}{T_{ij}^{old}}$. Figure 6.4 shows the effect of this transformation on the distribution of Question 2. The outliers on the upper side are concentrated around zero and the interviews with a fast response time stand out on the upper side, which makes it easier for the clustering algorithms to detect those interviews.

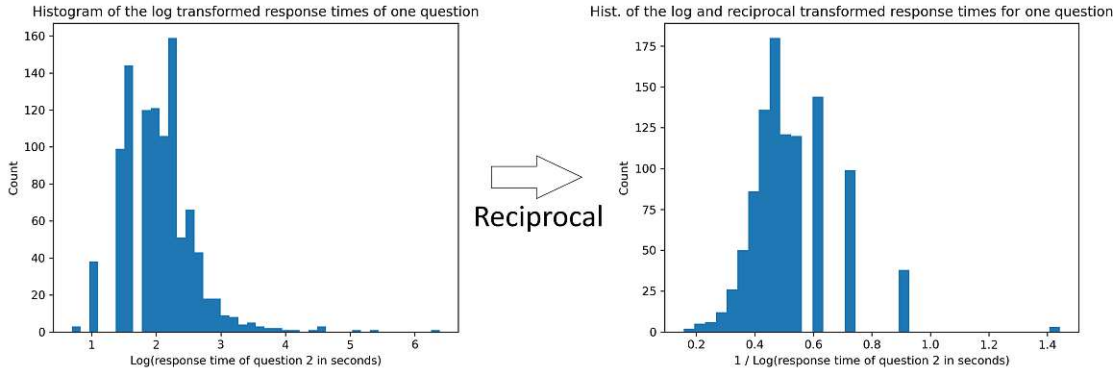

Figure 6.4: Effect of the reciprocal transformation on the log-transformed response times for Question 2.

Then, we do dimension reduction with standard scaling followed by principal component analysis (PCA). PCA dates back to Pearson (1901). The goal is to find components, which maximize the variance while achieving uncorrelated components, which corresponds to orthogonality in the framework of PCA. Equation 6.2 shows the initial matrix $X \in \mathbb{R}^{n \times p}$, which is transformed to the score matrix $Z \in \mathbb{R}^{n \times p}$ by the loadings matrix $\Gamma \in \mathbb{R}^{p \times p}$. To achieve dimension reduction, one typically uses not the whole Z matrix but a subset of the first q columns. The columns of Z are called principal components.

$$Z = X\Gamma \tag{6.2}$$

We want to find $\Gamma_{\cdot 1}$ such that $Var(Z_{\cdot 1})$ is maximized with the restriction that $||\Gamma_{\cdot 1}|| = 1$. This can be solved by Lagrange optimization. Then, we want to find $\Gamma_{\cdot 2}$ such that $Var(Z_{\cdot 2})$ is maximized with the restriction that $||\Gamma_{\cdot 2}|| = 1$ and that $Z_{\cdot 1}$ and $Z_{\cdot 2}$ are uncorrelated, which can be shown to be the same as orthogonal in this context. Again, we solve this with Lagrange optimization and continue doing this procedure with the restriction that $Z_{\cdot i}$ is uncorrelated to $Z_{\cdot j}$ for all $i \neq j$ until we get the whole Z matrix. By solving the optimization problems, one realizes that Γ can be optained by the eigenvalue decomposition of the covariance matrix of X.

PCA is a common tool and Read et al. (2021) use it while following a similar approach clustering the response times on a question level. They use a Gaussian mixture model and expectation maximization (EM) to cluster the principal components explaining 80% of the total variance. One can either choose the number of principal components based on a predefined threshold like 80% or one can inspect the scree plot shown on the left-hand side of Figure 6.5. This plot usually has a knee, and the number of principal components is selected based on the point where the remaining components exhibit a linear relationship. However, in Figure 6.5, it is not entirely clear where this point lies. There is a total of 32 variables ($p = 32$) and therefore a maximum number of 32 principal

components, which would explain 100% of the variance. Since we want a dimension reduction and that a fair amount of the total variance is explained, we select the first eleven principal components, which explain around 80% of the total variance in this case.

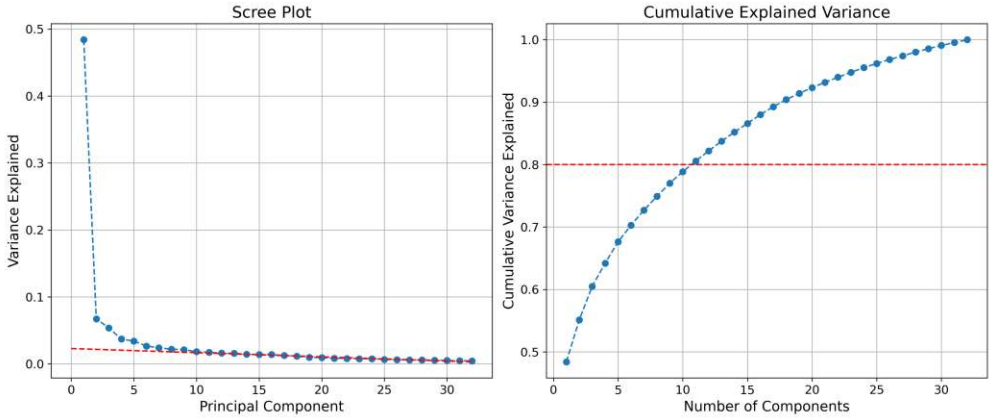

Figure 6.5: The left-hand side shows a scree plot indicating how much variance is explained by every principal component. The right-hand side shows the cumulative explained variance if this number of principal components is chosen.

We use PCA, because we want to remove noise from the data, which might negatively impact the performance of clustering algorithms. It is good practice to do standard scaling for every feature before applying PCA, because otherwise different scales of features can influence the variance maximization, which is not desired. Standard scaling transforms the data in a way that the expected value is zero and the variance is one like shown in Equation 6.3.

$$x_{new} = \frac{x_{old} - E[x_{old}]}{sd(x_{old})} \Rightarrow E[x_{new}] = 0 \text{ and } Var(x_{new}) = 1 \tag{6.3}$$

We standard scale the data and take the principal components explaining 80% of the total variance, i.e. choosing the smallest q such that

$$Var(Z_{\cdot 1}) + Var(Z_{\cdot 2}) + ... + Var(Z_{\cdot q}) > 0.8\ tr(Cov(X))$$

or decide the number of principal components based on the inspection of the scree plot.

6.3 Cluster algorithms and hyperparameters

The cluster algorithms have been explained in Section 2.2. Every cluster algorithm has hyperparameters, which have to be chosen in advance, e.g., the number of clusters for k-means. We use a variant of DBSCAN, where all noise points are again clustered with a k-means algorithm.

There are several ways to choose the ideal hyperparameters like random search or genetic optimization algorithms. The search space for our use case is not too big, because the hyperparameters are often integer values. Therefore, it is sufficient to do grid search, where every combination of hyperparameters based on a predefined search grid is tried. Table 6.3 shows the hyperparameters used for each algorithm during the grid-search. This is a common approach, e.g., done by Belete and Manjaiah (2021), who highlight the necessity of hyperparameter tuning to improve the performance of algorithms.

Algorithm	**Hyperparameters**
k-means	Number of Clusters = {2, 3, 4, 5, 6, 7, 8, 9}
Agglomerative Clustering	Number of Clusters = {2, 3, 4, 5, 6, 7, 8, 9}
DBSCAN with k-means	Epsilon = {0.1, 0.2, ..., 4.8, 4.9} Minimum Samples = {5, 6, 7, 8, 9} k-means number of clusters = {2,3}
Birch	Number of Clusters = {2, 3, 4, 5, 6, 7, 8, 9}

Table 6.3: Grid search hyperparameters for all clustering algorithms.

6.4 Silhouette score

It is not easy to decide which algorithm should be chosen since we are dealing with unlabelled data. A criterion, which was introduced by Rousseeuw (1987), is the silhouette score or average silhouette width. We choose the final algorithm and hyperparameter settings based on this score. Other possibilities would be the Calinski-Harabasz index, the Hartigan index or the Gap statistic.

Essentially, a score is computed for every observation in the dataset indicating whether the observation is well classified or not. We explain the computation of the silhouette score in the following.

We use an observation $X_{i\cdot}$ classified to cluster C_k. The average similarity/dissimilarity of the observation belonging to the classified cluster C_k is

$$d_{X_{i\cdot},C_k} = \frac{1}{n_k - 1} \sum_{j:X_{j\cdot}\in C_k, i\neq j} d^2(X_{i\cdot}, Xj\cdot).$$

Here, n_k denotes the number of observations in cluster C_k. The distance measure is denoted by d. This can be the Euclidean distance for example.

We want to compare the similarity of every observation to its own cluster to the similarity of this observation to its closest cluster. Therefore, we compute the similarity to all other clusters C_l, to which the observation $X_{i\cdot}$ does not belong.

$$d_{X_{i\cdot},C_l} = \frac{1}{n_l} \sum_{j:X_{j\cdot}\in C_l} d^2(X_{i\cdot}, X_{j\cdot})$$

We take the minimum value of the similarity to the other clusters.

$$d_{X_{i\cdot},C} = \min_l d_{X_{i\cdot},C_l}.$$

The silhouette value can be computed as

$$s_{X_{i\cdot}} = \frac{d_{X_{i\cdot},C} - d_{X_{i\cdot},C_k}}{\max(d_{X_{i\cdot},C}, d_{X_{i\cdot},C_k})}.$$

The value of $s_{X_{i\cdot}}$ is between -1 and 1, where a value close to 1 indicates that the classification to the cluster C_k was correct, whereas a negative value indicates that the observation was misclassified. We compute these silhouette value $s_{X_{i\cdot}}$ for every observation $i = 1, ...n$ and take the average. Finally, this gives us the silhouette score.

$$S = \frac{1}{n}\sum_{i=1}^{n} s_{X_{i\cdot}}$$

The silhouette score also yields a number between -1 and 1, where high values indicate good clustering performance.

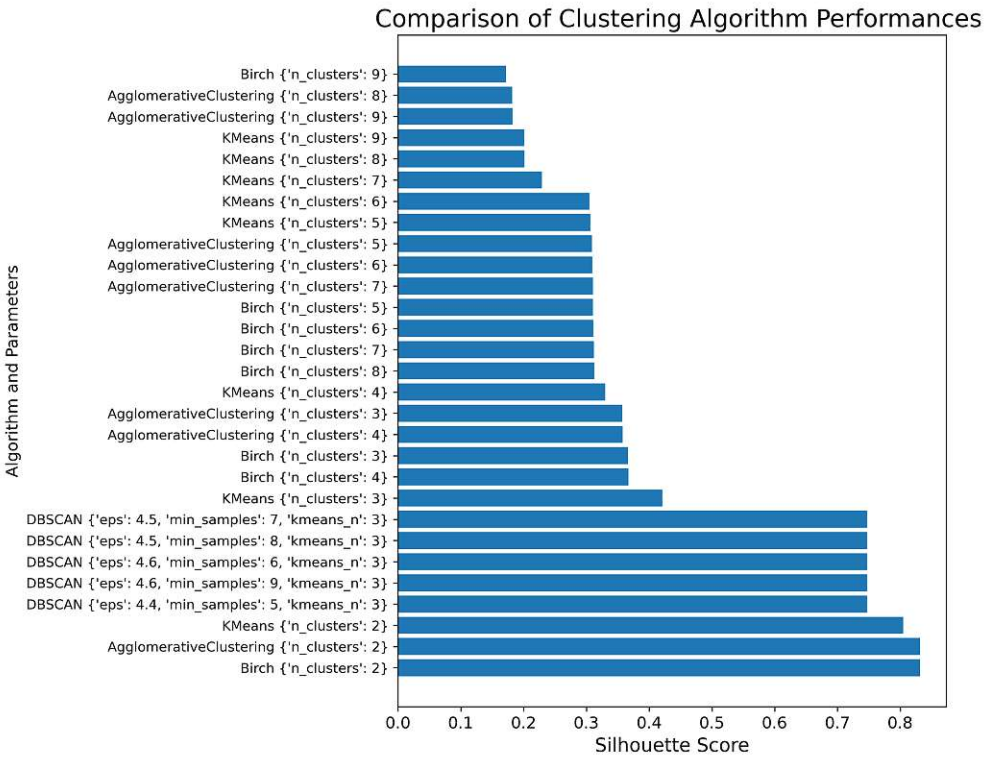

Figure 6.6: Silhouette scores for some algorithms with the predefined parameter settings.

Figure 6.6 shows the silhouette scores for all algorithms and parameter settings besides DBSCAN with k-means. For DBSCAN with k-means we only show the parameter settings with the five best silhouette scores. Based on Figure 6.6, we opt for the Agglomerative Clustering with the number of clusters equal to two, which achieves the same score as Birch clustering with the number of clusters equal to two, which can also be chosen. The "best" algorithm could differ for another dataset.

Figure 6.7 shows the first two principal components, which explain around 60% of the total variance, and the cluster labels obtained by Agglomerative Clustering. There is one main group, with some observations that do not align with it. Some of these "outliers" form Cluster 1.

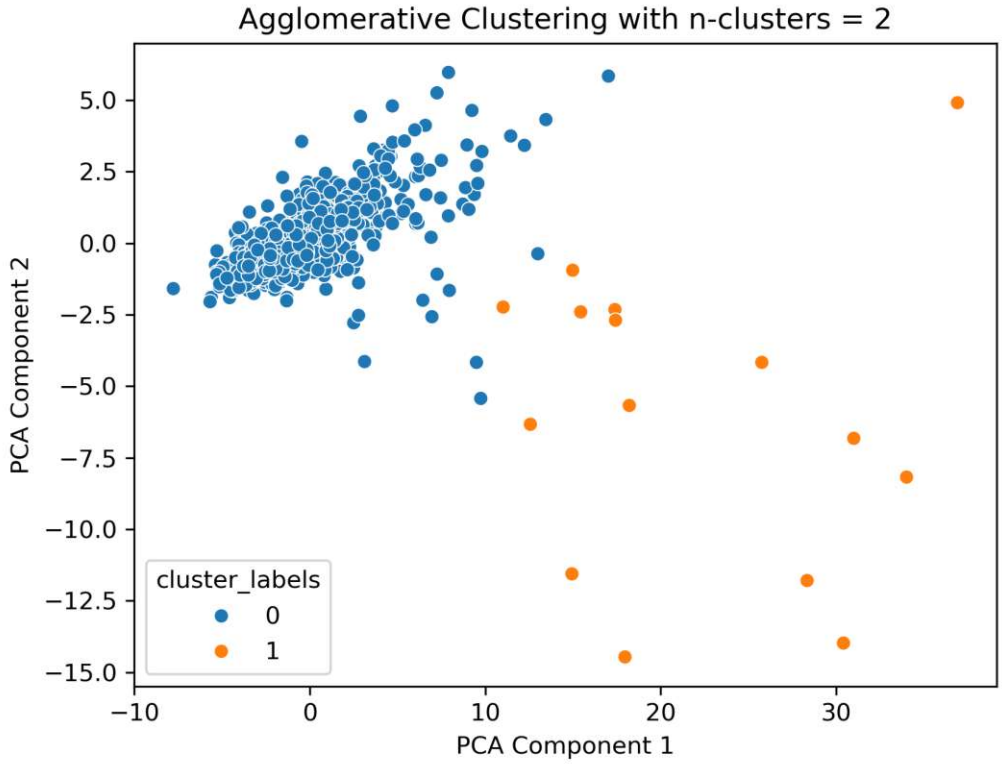

Figure 6.7: First two principal components with the cluster labels obtained by Agglomerative Clustering with the predefined number of clusters equal to two.

6.5 Flagging of interviews

Finally, we want to flag some observations based on the clustering. A common threshold to exclude interviews in online surveys is if the median of the total interview time is below 0.4 times the median of the total interview time of all interviews. Greszki et al. (2015)

for example use this threshold alongside other similar thresholds to exclude "speeders". We also use this threshold, but we do not flag single observations. We are flagging whole clusters if the median total interview time of the cluster is below this predefined threshold. Figure 6.8 shows boxplots of the log-transformed total interview times. The red line indicates the threshold and we see that the median of Cluster 1 is below the threshold. Therefore, all observations in this cluster get a flag for the time component.

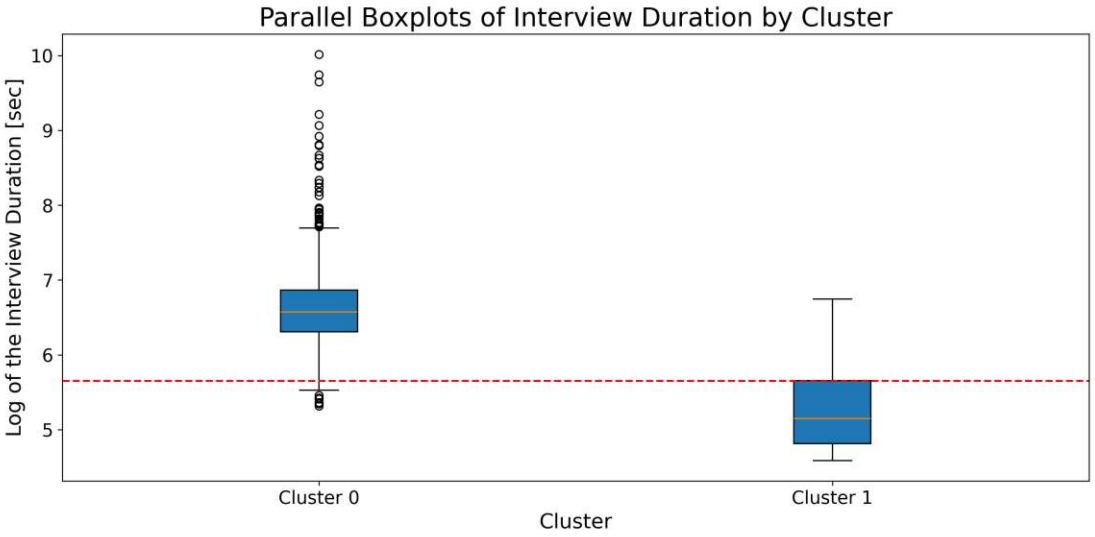

Figure 6.8: Parallel boxplots of the log-transformed total interview time of all clusters with a threshold for flagging clusters.

In our analysis, it's important to highlight that there are observations in Cluster 0 that fall below the threshold and observations in Cluster 1 that exceed the threshold. This pattern arises because the fast respondents in Cluster 0 exhibit a similar overall answering behaviour to other members of their cluster across different questions. They may take slightly longer on some questions, but they generally complete the survey faster, often because they are younger individuals using smartphones.

Conversely, respondents in Cluster 1 who exceed the threshold also display answering behaviour similar to other Cluster 1 members. They tend to be fast or very fast on most questions but take much longer on a few, resulting in a reasonable overall completion time. Our strategy effectively identifies these respondents, ensuring that people cannot manipulate speed tests by employing tactics like stalling on certain questions.

There are 15 observations in Cluster 1 which get a flag for the time component.

In our example there are only two clusters, but the hyperparameter setting could also be chosen to get more clusters. In this case, there is also the possibility that more than one cluster get a flag. Then, all observations in these clusters get a flag for the time component. This concludes the analysis of the response times.

CHAPTER 7

Answer patterns

Alongside the response times, the answer pattern is the other component, which we use to preselect useful interviews. We assume that bad respondents have a different answer pattern than the majority of the interviewed people. This is due to the fact that they answer inattentive or without focussing on the question contents, which leads to more or less random responding or selecting the same answer several times. The difference in the answer pattern can be seen as an anomaly and an autoencoder should detect these anomalies and allow the flagging of these respondents. To achieve this goal, we perform different steps, which are described in this chapter.

7.1 Preprocessing

Preprocessing is necassary to get an input which can be used by an autoencoder. We have to handle filter questions in order to use all available information. The input for an autoencoder has to be a numeric vector, which also needs to be ensured by preprocessing.

7.1.1 Handling of filter questions

There can be filter questions, which are not shown to every participant, which leads to missing values. There are a few options to handle these missings. One would be to just completely leave these questions out, but we want to use all available information. Therefore, we keep the questions and introduce a new answer category indicating that the question was not asked to this participant. Table 7.1 and Table 7.2 illustrate the concept with an example.

ID	Do you have a pet?	Is your pet allowed to sleep in your bed?
1	Yes	Yes
2	Yes	No
3	No	

Table 7.1: Survey data before handling missing values.

ID	Do you have a pet?	Is your pet allowed to sleep in your bed?
1	Yes	Yes
2	Yes	No
3	No	Not Applicable

Table 7.2: Survey data after handling missing values. "Not Applicable" indicates that the question was not shown to the participant due to a previous filter question.

7.1.2 Partial interviews

We offer a strategy to include partial interviews in the analysis. This can be seen as a form of data imputation. Typically, data imputation is done in case of the non-response of an item, which means that somebody chooses "no answer" for a question. There are several strategies to impute those missings discussed in the literature, e.g. by Nordholt (1998), Rubin and Schenker (1991) or Pratesi and Salvati (2006).

We do not impute the non-response items for the implausible interview detection since this is an important information for us. However, if participants who did not finish the full survey should be included in the analysis, then we rely on a simple and well known data imputation technique. We draw the value for the imputation randomly from similar respondents based on a certain attribute like the age group. This is sufficient for our purpose.

7.1.3 Data transformations

We deal mostly with closed questions, where people can select from a predefined set of answers. In this case, the answer variable can be either on a nominal or ordinal scale. If there are open questions, where people have to answer with natural language, we cannot include these questions. However, a separate analysis of questions with natural language as answers can be insightful. Especially if the answers are in gibberish or formulated too well, this can indicate insincere answer behaviour. Open questions with numeric values as required answers are feasible, e.g. questions like: "What is your age?" or "What is your income in Euros?".

Nominal variables

Nominal variables are categorical variables, which cannot be sorted. E.g., the question "What is the color of your eyes?" has nominal answers, because the colors cannot be

sorted. However, to use this information for our model, one has to transform the answers to numeric values. This is typically done with one-hot encoding. Tables 7.3 and 7.4 show the one-hot encoding of an example. Every categorical value corresponds to a column with binary values, where only the column with the original value is 1 while the other columns are 0.

Person	**Eye Color**
Person 1	Blue
Person 2	Green
Person 3	Brown
Person 4	Blue

Table 7.3: Example data before one-hot encoding.

Person	**Blue**	**Brown**	**Green**
Person 1	1	0	0
Person 2	0	0	1
Person 3	0	1	0
Person 4	1	0	0

Table 7.4: Example data after one-hot encoding.

Ordinal variables

Ordinal variables are also categorical variables, but it is possible to sort the categories. An example is the question "How satisfied are you with your health condition?" with the possible answers "Very satisfied", "Rather Satisfied", "Rather unsatisfied", "Very unsatisfied". For ordinal variables, it is possible to do one-hot encoding as well, but one could also map the answers to a scale and use just one column with the mapped values. The mapped values could be just 1,2,3,4 assuming the distance between the possible answers is equal. This depends on the question, but for questions on a Likert scale, this mapping is usually feasible.

Open questions

As already mentioned, we can only include numeric open questions like "What is your age?". Open questions with natural language as required answer cannot be used for our model.

As final input for our model, we use all feasible questions and preprocess them based on the data type like described above. Then, we perform standard scaling for every column. This is done to avoid different scales, which might have a negative impact on the model performance. Equation 7.1 describes the standard scaling, which leads to a mean of zero and a standard deviation of one for every column.

$$x_{new} = \frac{x_{old} - E[x_{old}]}{sd(x_{old})} \Rightarrow E[x_{new}] = 0 \text{ and } Var(x_{new}) = 1 \tag{7.1}$$

This way we get a data matrix $X \in \mathbb{R}^{n \times p}$ with only numeric columns, where n is the number of participants and p the number of features derived from the questions after preprocessing.

7.2 Autoencoder

We use an autoencoder, which is described in Section 2.3, to detect anomalies in the answer patterns. The hyperparameters for the model are crucial for its performance. We explain the choice of hyperparameters and the calculation of the reconstruction error with the use of cross-validation.

7.2.1 Hyperparameters

We want to provide a general approach to preselect useful interviews. Therefore, we must take into account that the number of survey columns after preprocessing (p) and the number of interviews (n) differ across surveys. The hyperparameters should be automatically adapted based on these two parameters. In Section 2.3 the key-hyperparameters of an autoencoder are described. For most machine learning tasks, one chooses the hyperparameters in a way that the MSE on the validation set is minimal. When doing anomaly detection, this is not always the desired goal since a minimal MSE might lead to the model incorporating features describing outliers. Outliers or anomalies might then be classified as "normal" observations. This could happen if the number of nodes and layers is too big, but if there are not enough nodes and layers the model might not be able to learn useful features at all. Sabetti and Heijmans (2021) also discuss this topic of hyperparameter search for anomaly detection with autoencoders. However, they have the advantage to have a training set without anomalies, which makes it feasible to minimize the MSE.

We have to train an autoencoder for every new survey and do not want to do hyperparameter optimization every time since this can be very time-consuming. For this reason, we provide a strategy, which might not necessarily lead to a minimal MSE, but to a difference in the distribution of the reconstruction error between normal observations and anomalies.

The number of survey columns after preprocessing (p), which is usually much higher than the initial number of survey questions due to one-hot encoding, and the number of interviews (n) are used to adapt the hyperparameters. We use the following strategy for the selection of the hyperparameters:

- **Nodes per Layer**: The number of nodes per layer starts at the input dimension p and is reduced by a specified reduction factor r. The first layer after the input

layer has the dimension $p \cdot \frac{1}{r}$. The following layer has the dimension $p \cdot \frac{1}{r^2}$ and so on until the predefined minimum layer size, e.g. $\frac{p}{10}$, is reached. Then, the dimension is increased in the same way until it reaches the input dimension again. This approach creates a pyramid-shaped architecture which is effective for encoding complex patterns in the data.

- **Number of Layers**: The number of layers is determined dynamically based on the input dimension and a minimum layer size. Layers are added until the reduction factor brings the size below the minimum layer size as described above. This ensures the model is neither too simple nor too complex for the data.

- **Learning Rate**: We use a learning rate range test to select the optimal learning rate for training the autoencoder. This method was introduced by L. N. Smith (2017) and involves exponentially increasing the learning rate over a specified range and recording the corresponding loss. In our case, we vary the learning rate between 0.0001 and 1. The optimal learning rate is then selected based on the point where the loss starts to decrease most rapidly. This approach ensures that the learning rate is chosen to facilitate efficient training while avoiding instability.

- **Batch Size**: The batch size is determined as a function of the number of observations, ensuring that it is neither too small (which would make training noisy) nor too large (which would make training slow). Specifically, it is set to a maximum of 32 or one percent of the total number of observations.

- **Activation Function**: The ReLU (Rectified Linear Unit) activation function is used for all hidden layers. This choice is motivated by its simplicity and effectiveness in preventing vanishing gradients, which is essential for training deep networks.

- **Epochs**: The number of epochs is capped at 500, with early stopping applied based on the validation loss. Early stopping with a patience of 10 epochs helps to prevent overfitting by stopping training when the validation loss stops improving.

We suggest values for the reduction factor r and minimum layer size/bottleneck layer size in order to provide a completely automated method. We recommend the value 3 for r and the value $\frac{p}{10}$ for the minimum layer size. For example, if p is equal to 600, the layers have the size 600-200-67-22-67-200-600. These recommendations are based on experiments with different surveys. The values lead to a separation of the distributions of interviews with high reconstruction errors and low reconstructions errors in all experiments. It would also be possible to inspect the distribution of the reconstruction errors and choose all hyperparameters based on the separation individually for every survey.

7.2.2 Calculate reconstruction errors with cross-validation

The goal is to calculate the reconstruction error for each respondent in the survey. The training should be done in a way that every observation is not included in the training

process to obtain the model used to calculate its own reconstruction error. This is achieved with k-fold cross-validation illustrated in Figure 7.1. The dataset is split into k equal sized parts and all parts but one are used for the training of the autoencoder. This gives us a model which can be used to calculate the reconstruction errors for the observations in the part of the dataset which was not used. Then, another part is left out and the model is trained with the remaining parts until every part was left out once and we get the reconstruction errors for every observation in the dataset.

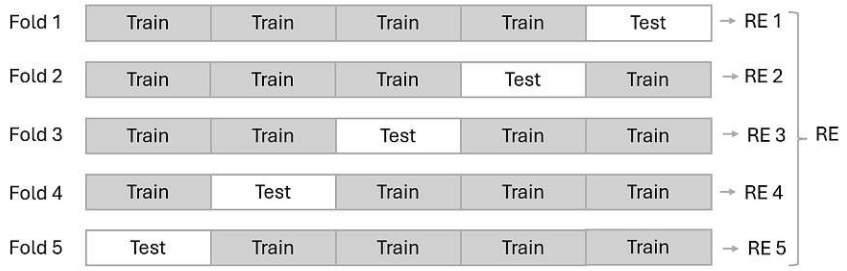

Figure 7.1: Illustration of k-fold cross-validation with the number of folds $k = 5$.

We already have the flags for the time component, and we assume that fast respondents are also more likely to have high reconstruction errors. Therefore, we use this information and perform stratified cross-validation. This is a well-known machine learning approach to ensure that each fold is representative of the whole dataset in terms of class proportions (Kohavi (1995)). For our case, this means that the number of "speeders" should be the same for every fold.

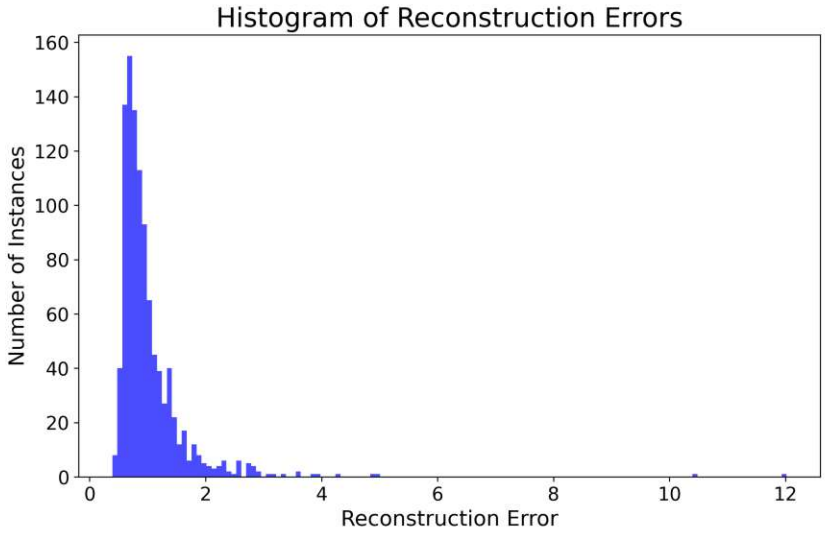

Figure 7.2: Histogram of reconstruction errors.

We use stratified 5-fold cross-validation, where we use the time-flags for stratification. Figure 7.2 shows the distribution of the reconstruction errors obtained through this cross-validation. It suggests that the observations can be divided into two different groups. The majority of reconstruction errors seems to be almost normally distributed, whereas a smaller group seems to have higher reconstruction errors.

7.3 Flagging of interviews

We want to flag suspicious interviews based on the answer patterns as we did it with the response times. We use the reconstruction errors to do this. A high reconstruction error indicates that there is an anomaly in the answer pattern of this respondent. It is crucial to keep in mind that an anomaly per se is not a bad thing because it might be possible that one person just gives completely different answers than the rest. Therefore, it does not necessarily mean that an interview has to be excluded if it got a flag. However, also random responding or choosing always the first or last answer might lead to an anomaly, which we want to detect. Figure 7.3 shows the distribution of the reconstruction errors with the inclusion label based on other checks. We see that excluded interviews are more likely to have a high reconstruction error.

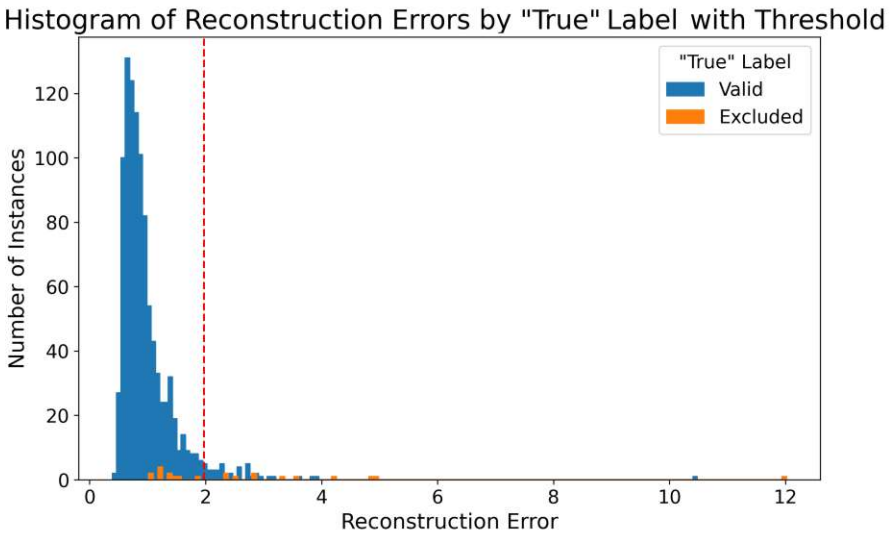

Figure 7.3: Histogram of reconstruction errors with inclusion/exclusion label based on other checks and the 5% threshold (1.97).

Additionally, Figure 7.3 shows the 5% threshold. The 5% of interviews with the highest reconstruction errors are above this threshold and get a flag for the answer component. The threshold should separate the two underlying distributions which is in our case achieved by the predefined 5% threshold. There would be also the possibility to inspect the distribution of the reconstruction errors and define a threshold based on this.

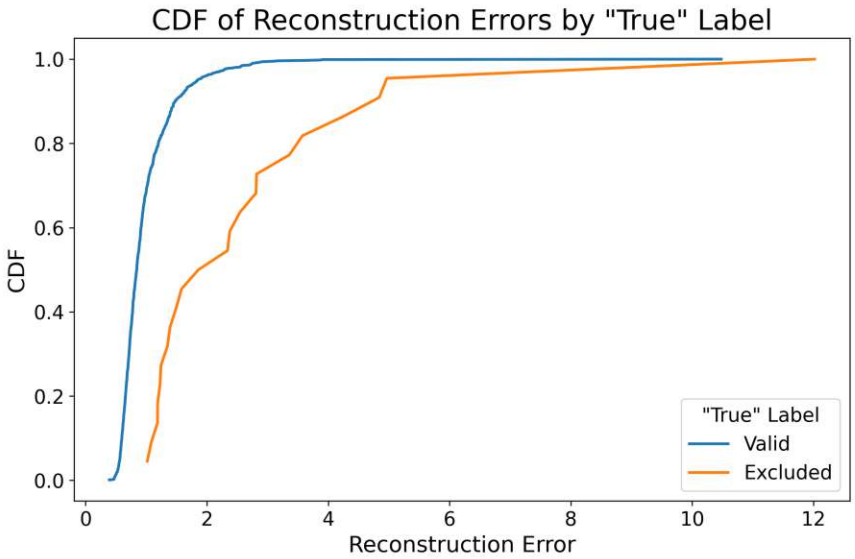

Figure 7.4: Empirical cumulative distribution function (CDF) of the reconstruction errors with inclusion/exclusion label based on other checks.

The inclusion/exclusion labels based on other checks also allow us to compare the reconstruction errors of the "Valid" and "Excluded" group based on other checks. The empirical cumulative distribution function (CDF) of the reconstruction errors is shown in Figure 7.4. The empirical CDF indicates the percent of interviews of each group, which are below the given point. For better understanding, we give a short example. We look on the x-axis at the reconstruction error value 2. The value of the empirical CDF of the interviews with "Valid" label is almost at 1, which means that almost 100% of interviews with the "Valid" label have a reconstruction error below 2. The value of the empirical CDF of the interviews with "Excluded" label is around 0.5, which means that around 50% of interviews with the "Excluded" label have a reconstruction error below 2, which also means that around 50% have a value above 2. This clearly indicates that the implausible interviews based on other checks tend to have higher reconstruction errors than the plausible interviews.

To get a better understanding how the autoencoder works and which interviews get a flag, we manually inspect some of the interviews with the highest reconstruction errors (RE). This is done in Table 7.5. The inspection shows that the autoencoder is capable of detecting different kinds of bad answer behaviours if the majority of the other respondents has a "normal" answer behaviour. The autoencoder recognizes significant outliers for questions where participants are asked to enter a number, answers with logical inconsistencies between different questions and a high amount of the selection of "no answer".

RE	"True" label	Inspection
12.0	Excluded	The participant answered the question "Please enter your age." with 7070, which obviously cannot be true and of course is an outlier. This is the main reason for the high reconstruction error since most other questions seem to have nothing unusual besides the fact that the person answered a filter question, which was only asked to people not born in Austria. This is not an attribute which should lead to exclusion but still might increase the reconstruction error since only a few people, who answered the questionnaire, were not born in Austria.
10.5	Included	The participant answered the question "How many square meters of living space are available to you/your household?" with 4527, which is a very big outlier, especially since the person stated to live in Vienna and to be unemployed in other questions. There were no obvious anomalies other than that.
5.0	Excluded	The participant answered an unreasonably high number of questions with "No answer".
4.8	Excluded	The participant answered a lot of questions with "No answer". There was a question like "How easy or hard is it for you?". Then there were different things listed and the person selected "No answer" for all twelve of these things and also for a lot of other similar questions.
4.2	Excluded	The participant answered nearly every question with "No answer".
3.9	Included	The participant chose very extreme answers, like "always" and "never", which leads to the high reconstruction error. Nevertheless, the answers make sense, and this is an example that a high reconstruction error can also mean that it is just an anomaly, but not a bad respondent.
3.9	Included	The participant answered quite a few questions with "No answer". Additionally, there were some discrepancies between questions. For example, the person answered the question "How is your health in general? " with "Good" but also answered the question "How much have you been limited in activities of daily living due to a health problem for at least six months? Would you say you are:" with "Severely limited". These answers kind of contradict each other and also the autoencoder might have learned another relationship between these two questions.

Table 7.5: Manual inspection of the 5 interviews with the highest reconstruction errors

Additionally, there is also an example of a respondent who gets a flag of the autoencoder but should not be excluded because the answers are unusual and extreme but not necessarily bad. This highlights the importance to not just blindly eliminate all the questions flagged by the autoencoder.

In our example, all interviews with a higher reconstruction error than the threshold of 1.97 (5% threshold) get a flag for the answer pattern component. This concludes the analysis of the answer patterns.

CHAPTER 8

Results

We apply the proposed method on the two datasets and evaluate the results. Additionally, we perform statistical analysis on the effect of not excluding bad respondents.

8.1 Dataset 1

Dataset 1 is a real world survey for the Gesundheit Österreich GmbH described in Section 5.1.1. We calculate the confusion matrix obtained with the comparison of our strategy and other checks based on the expertise of the company OGM research & communication GmbH. There are four different outcomes: no flag, a flag for one of the two components or a flag for both components. Table 8.1 shows that only two observations, which do not get a flag, are excluded based on other checks. We also see that one component would not be enough since we would miss quite a high number of interviews.

Other checks	**No Flag**	**Time Flag**	**Answer Flag**	**Flag for both**
In	964	1	38	3
Out	2	9	9	2

Table 8.1: Confusion matrix for all different outcomes.

It is not a surprise that there is an overlap between the time component and the other checks since speed checks have also been done in the underlying elimination process. However, it is a little bit more surprising that the autoencoder is indeed capable of detecting strange answer patterns like often "no answer" even though overall a quite high number of interviews are flagged for the time component.

Table 8.2 only differentiates between flag or no flag. This matrix can be used to calculate evaluation metrics like accuracy (95.7%) or recall (90.9%).

Other checks	No Flag	Flagged
In	964	42
Out	2	20

Table 8.2: Confusion Matrix for flag/no flag.

The goal is to preselect useful interviews and we are able to detect over 90% of the bad interviews. Further checks or manual inspection is needed for a final selection.

The misclassified examples, especially those, which got no flag or a flag for both, are of amplified interest since we want to find out what the reason for the misclassification was. To do this, we manually inspect and analyze those five interviews.

8.1.1 Interviews: both of our flags but included by other checks

We start with the interviews which got a flag by our time and answer component but would be included based on other checks. These are three interviews:

Interview 1

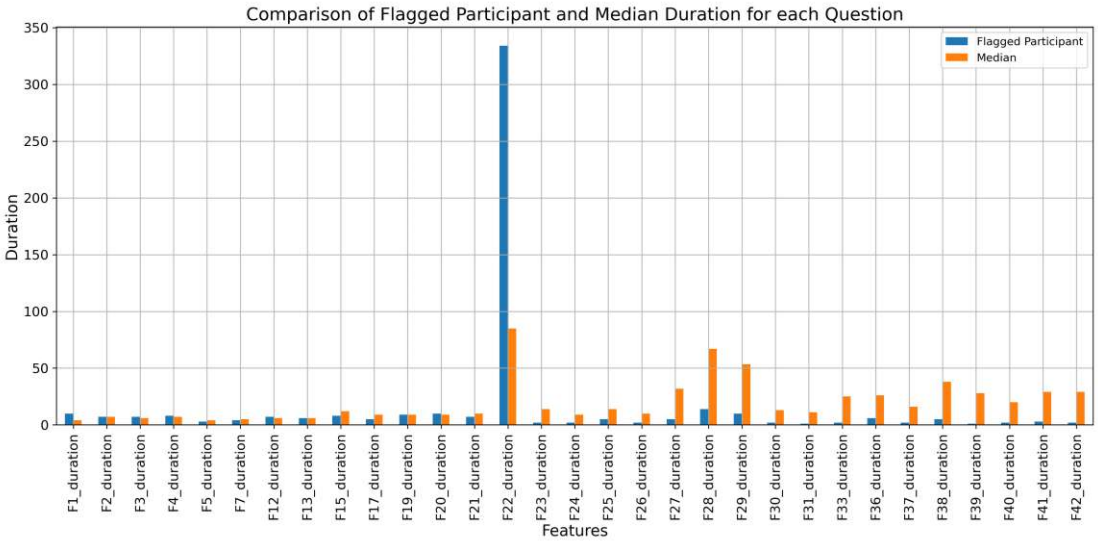

Figure 8.1: Comparison of the median answer time and the answer time of the respondent of interest for every question.

We look at the times on a question level and compare the times with the median answer time for every question. Figure 8.1 shows that for the first few questions the time was around the median of these questions. The participant took really long for question "F22", which is a question where also the median is higher than for all other questions, but then finished the survey in really short time. The answer times for those questions are way faster than the median and it would not be possible to answer the questions in a faithful

way in such an amount of time. This is also the reason why this observation is a part of the " too fast" cluster because the later questions are a clear sign of speeding. A reason might be that the questionnaire was too long for the respondent and the motivation dropped after question "F22". Then, the respondent might have just clicked through the rest of the questions as fast as possible. This example shows why the question level of the response times is necessary. The overall response time would have not been suspicious due to the long time taken for question "F22".

The manual inspection of the patterns does not show many clear signs of suspicious behaviour. There is only obvious inconsistency: the question about the overall health condition was answered with "Good" and the satisfaction with the own health was rated with 2 out of 10 (0 = not satisfied at all, 10 = completely satisifed), which indicates implausibility. One can additionally assume that the questions from "F23" onward were not answered faithfully and since also the autoencoder flagged this interview, there are some other anomalies in the answers as well. Overall, the exclusion of this interview would definitely be justified.

Interview 2

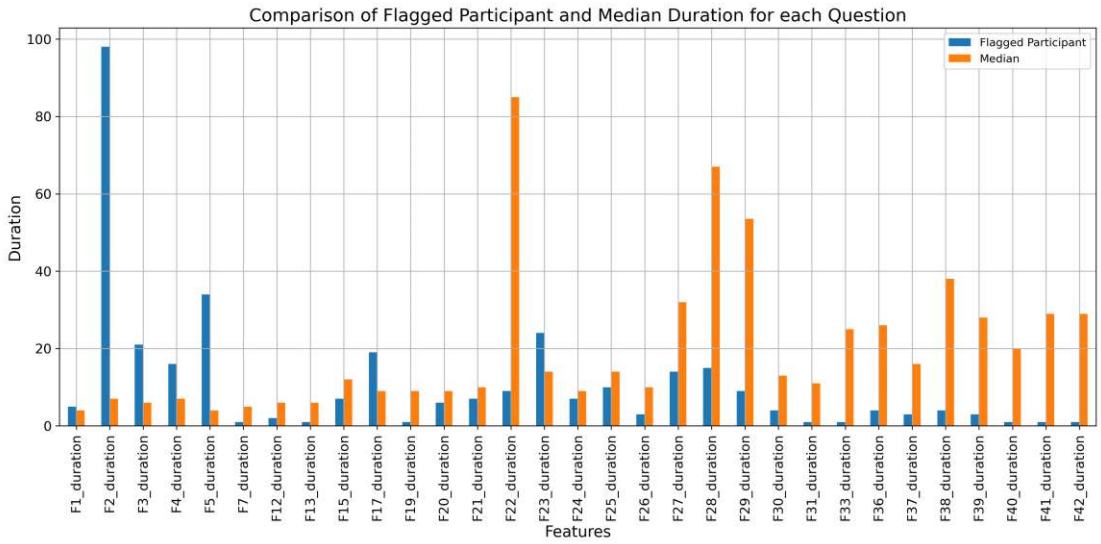

Figure 8.2: Comparison of the median answer time and the answer time of the respondent of interest for every question.

The question level times of Interview 2 are shown in Figure 8.2 in comparison with the median times. The respondent takes longer than the median for the first five questions but then continues to answer the majority of questions in a too short amount of time. Again, this is the reason why the respondent is classified in the "speeder" cluster even though the overall completion time is reasonable.

By looking at the answers manually, there are no obvious abnormalities, but the flag

of the autoencoder indicates an anomaly, which alone would not be decisive but in combination with the fast responses calls for the elimination of the interview.

Interview 3

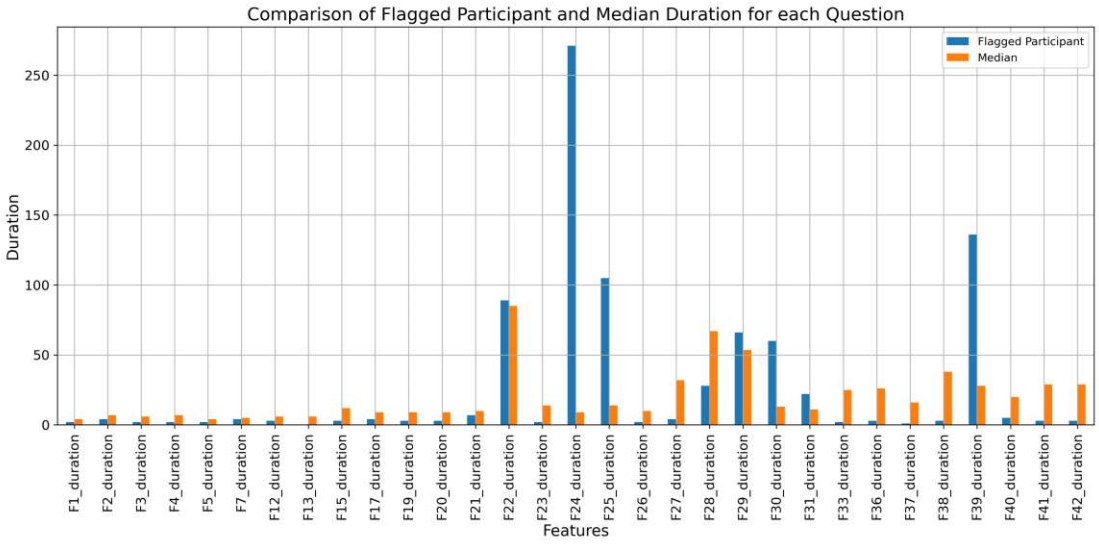

Figure 8.3: Comparison of the median answer time and the answer time of the respondent of interest for every question.

As for the other two interviews, the respondent of Interview 3 is also too fast for most questions. For some questions, the respondent took way longer than the median of all respondents for these questions leading again to a reasonable total completion time.

The manual inspection of the answers does not show obvious implausibility. However, the flag of the autoencoder is an indicator of suspicious answer behaviour. Therefore, it would be fine to eliminate this interview.

Findings

Overall, the three interviews clearly show how superior the analysis of the answer times on a question level is compared to the analysis of the total answer time. The number of questions with too fast answer behaviour would already justify an exclusion of the interviews. Even though the manual inspection does not always indicate clear signs of implausibility, the autoencoder flagged all three interviews indicating some kind of anomaly, what makes the justification of an exclusion even stronger.

8.1.2 Interviews: none of our flags but excluded by other checks

As already mentioned, the vast majority of interviews, which is excluded based on the other checks, is detected. However, there are also two interviews which were not flagged

by our method, but would be excluded based on other checks. This is arguably the case which should be avoided the most since we do not want to miss implausible interviews and therefore we inspect the two interviews to see what the reason for the exclusion might have been and why it was not detected by our method.

Interview 1

We start with the inspection of the response time on a question level. Figure 8.4 shows the comparison with the median response times on a question level. There is no indication of implausibility and also the overall response time is around the median.

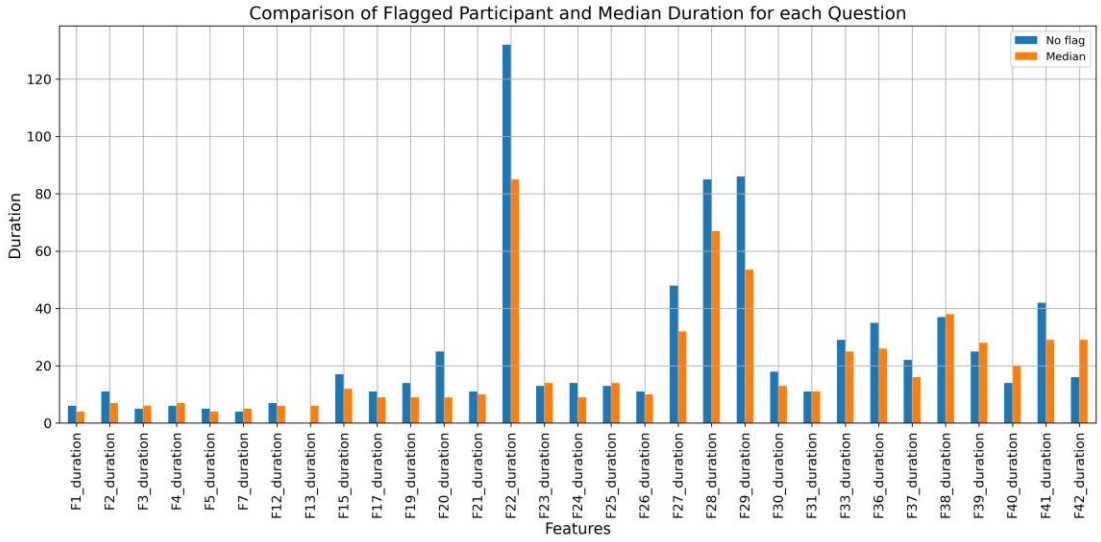

Figure 8.4: Comparison of the median answer time and the answer time of the respondent of interest for every question.

For this reason, we dig deeper and analyze the answer behaviour manually. The reason, why the other checks excluded this interview, is that "no answer" was frequently chosen for specific questions. However, the majority of questions was answered in a seemingly faithful way without choosing "no answer". There is non-response only for some sensitive questions. It seems that the respondent just refused to give a declaration for questions which the respondent feels uncomfortable to answer. If this would be the case for many respondents, this might indicate that these questions are not asked in the right way. The exclusion can be justified if someone answers with "no answer" too often, because the person might not be interested in the survey and just wants to finish it. This can especially be the case if there is some sort of incentive. Additionally, the respondent might not contribute any meaningful information, if there is no declaration for almost all questions. In our case, this does not seem to be the case, since the respondent has taken time to answer the questions and have also answered the majority of questions without selecting "no answer".

It is not completely clear if the interview should be excluded or not, because one can argue with the frequent use of "no answer" for an exclusion, but the overall picture with the response times and the sensitive questions which were not explicitly answered provide also arguments that this interview should not be eliminated. The inclusion is also what our method suggests.

Interview 2

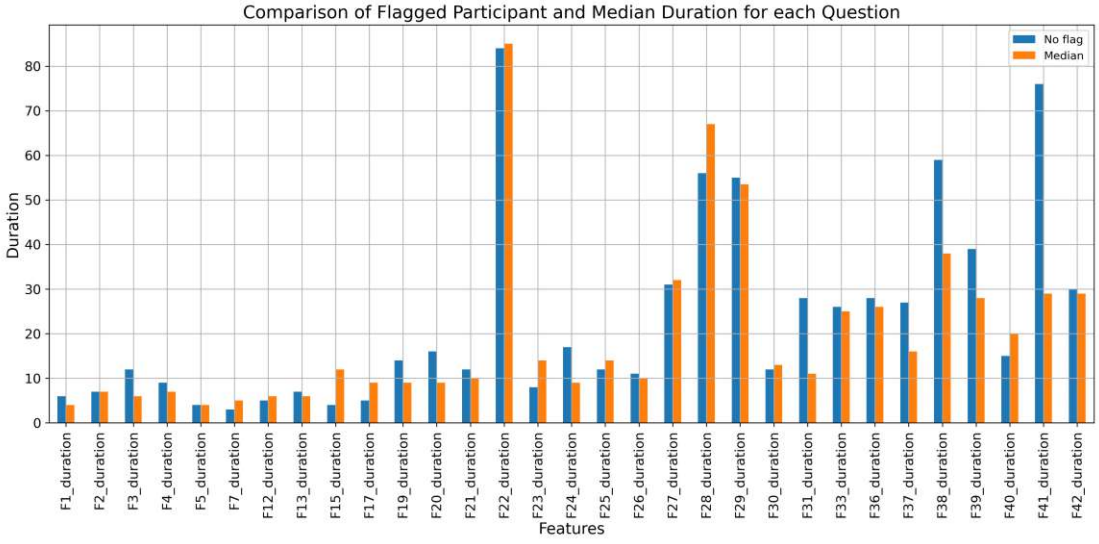

Figure 8.5: Comparison of the median answer time and the answer time of the respondent of interest for every question.

The analysis of the answer times for Interview 2 is shown in Figure 8.5. Again, there is nothing suspicious. The response times are mostly around the median, which is also true for the overall completion time.

The answer behaviour is also not that suspicious. "No answer" was chosen for some questions. This is the reason, why the other checks would exclude this interview. However, the choice of "no answer" is not excessive. Almost everything from the paragraphs from "Interview 1" about the inclusion and exclusion applies here, too. Another sign of faithful answer behaviour is that an optional open text question has been answered in a meaningful way. Therefore, an inclusion would be even more justified.

Findings

The two interviews, which get no flag from our method but would be excluded based on other checks, are both borderline cases. One can find arguments for the inclusion as well as for the exclusion of the interviews. This highlights the difficulty of the whole task, because even with manual inspection the desired label is not 100% clear. It is definitely not a tragedy to include these two interviews since the respondents do not seem to have answered randomly or unfaithfully, but just more frequently with "no answer".

By additionally considering the unsuspicious answer times, we tend more towards an inclusion of these two interviews.

8.1.3 Comparison to existing methods

We use evaluation metrics to compare our method with the most common existing methods. The methods have been described in Section 4.2. It is often recommended in the literature, e.g. by Curran (2016), to use a multiple hurdles approach. This means that several checks are applied, and the interview passes only if it passes a certain number of checks. We also do that in a way that we mark every interview if only one of the checks (longstring analysis, intra-range variability, antonyms and overall time) is not passed. The results are shown in Figure 8.6. We also show the single components of our detection strategy to see how the combined label is composed into the individual parts. Recall is the most important metric and our method rarely misses an interview as we saw before. The multiple hurdles approach comes the closest in terms of recall, but it is still worse and here the accuracy drops more than with our approach.

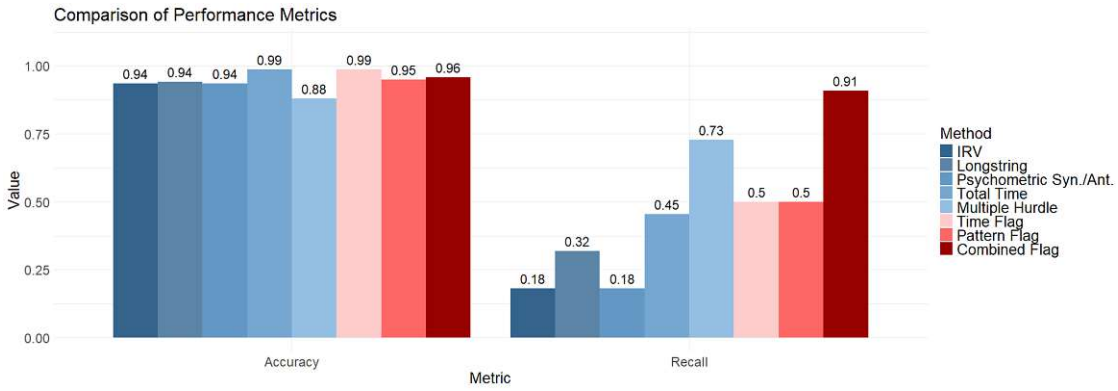

Figure 8.6: Evaluation metrics for different strategies to detect implausible interviews.

Not surprisingly, using only the time also performs quite well. This is true for our clustering of the times on a question level as well as for the detection strategy with the overall completion time. One obvious reason is that the "Total Time" method used is similar to the check, which is part of obtaining the labels. Using only the time allows for achieving a good accuracy, but the recall is only around 50%. The recall for only the answer component is also around 50%. This means in both cases that only the half of bad respondents are detected, whereas with the combined method we detect over 90% of bad respondents. This indicates that one should use both, the answer times and the answer patterns, to detect useful interviews. The combined strategy outperforms all other methods based on the recall, which is the most important metric, and still achieves one of the highest accuracies.

8.2 Dataset 2

Dataset 2 (Schroeders et al. (2022)) is described in Section 5.1.2. People were told to fill out the questionnaire either carelessly or faithfully. Unfortunately, there are only response times on a page level available. There are six pages in the questionnaire. The analysis of the response times shown in Figure 8.7 indicates that there are also respondents answering seemingly too fast without getting the instruction to answer carelessly. This is a sign that the labels are not 100% accurate. Additionally, only some careless respondents answer faster than the rest with a big fraction having similar answer times as the other respondents.

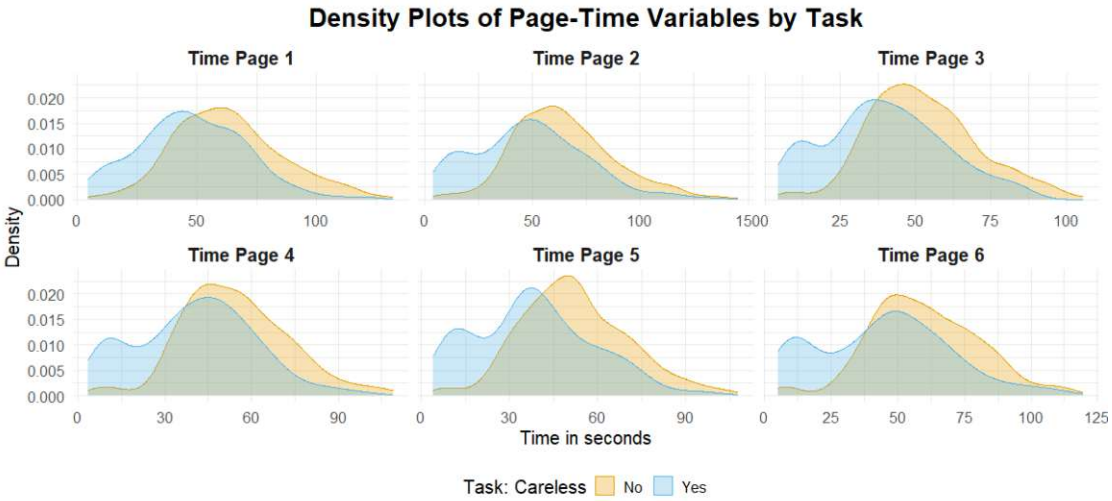

Figure 8.7: Comparison of the density of the answer times per page between people with the task of answering carelessly and the rest.

The authors use supervised machine learning to predict if a respondent answered carelessly or not. They create bootstrap samples and use different detection strategies for comparison. Thanks to the publication of the source code by the authors, we can use the same bootstrap samples and apply our detection strategy for comparison. In the composition of each bootstrap sample there are 162 "normal" respondents and 18 respondents who were told to answer carelessly. The limited number of observations in each sample makes it more difficult for the autoencoder to detect suspicious answer behaviour. Additionally, there are over 10% of implausible interviews, which should be detected, in every sample. The number of implausible interviews is typically lower in a closed survey environment. We apply our method as described in Chapter 6 and Chapter 7 with the only differences that not 5% but 10% of the interviews are flagged by the autoencoder and all PCA components are used for the answer times since the times are already aggregated on a page level.

Even though the authors use a supervised approach, they were not able to achieve the desired results in terms of evaluation metrics. This hints that the groups are hard to

separate. Figure 8.8 shows the accuracy of some traditional methods in blue tones (see 4.2), the proposed method by Schroeders et al. (2022) which is a Gradient Boosting Machine (GBM) using either the responses GBM_{Res}, the response times GBM_{RT} or both GBM_{Res+RT} in orange tones and our method using either one component or both components combined in red tones. The points in the plot are 100 randomly sampled observations out of the 1000 bootstrap samples and there is a boxplot of the accuracy of all bootstrap samples. The accuracy of our method using only the response times performs best.

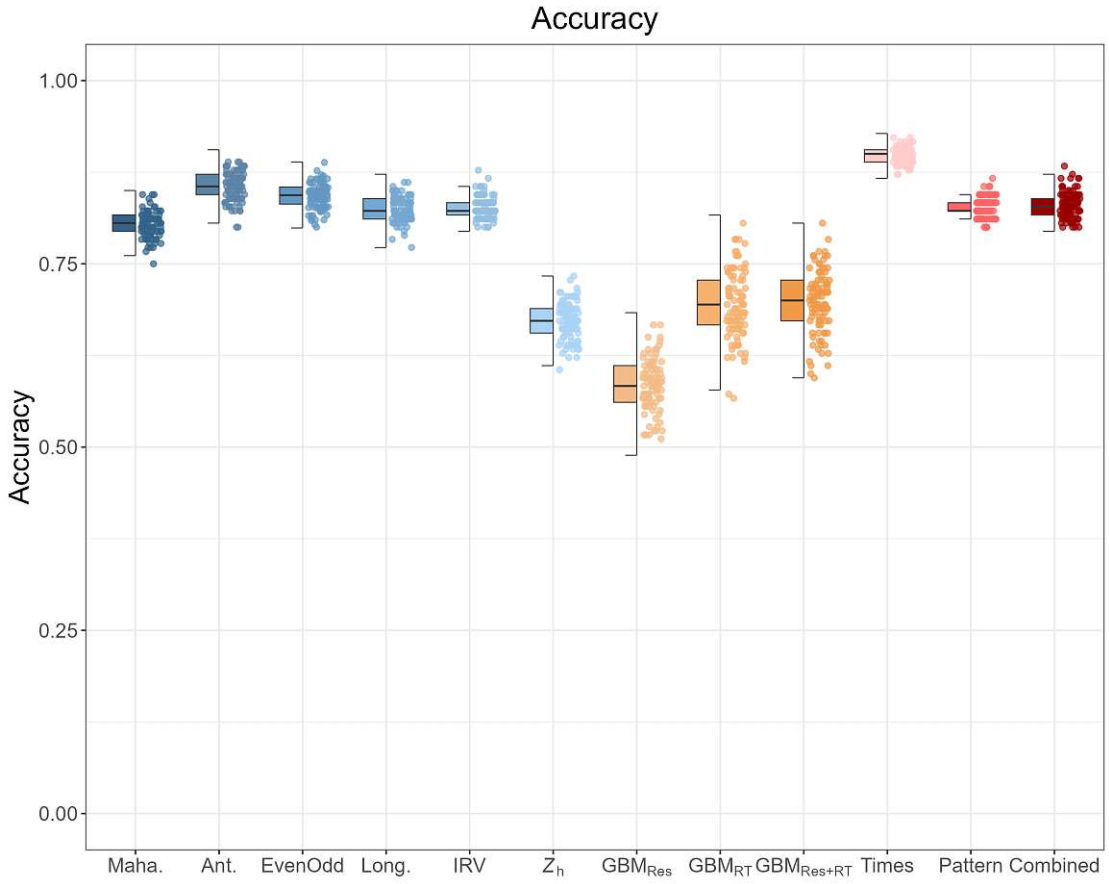

Figure 8.8: Comparison of the accuracy of different methods using Dataset 2.

At first glance, our method looks great compared to the others if we would only consider the accuracy. However, we have to remember that the recall, which is also called sensitivity, is the most important evaluation metric for our task. The sensitivity tells us the fraction of bad respondents detected. Here, it does not look so great anymore. This is visualized in Figure 8.9. All methods struggle to detect bad respondents and we also detect only around one fourth of bad respondents with the combined approach. Even though this is better than most traditional methods, the supervised approach and the Z_h

statistic achieve higher sensitivity scores. One reason is that the number of interviews flagged is higher for these methods. This also leads to the reduced accuracies in Figure 8.8.

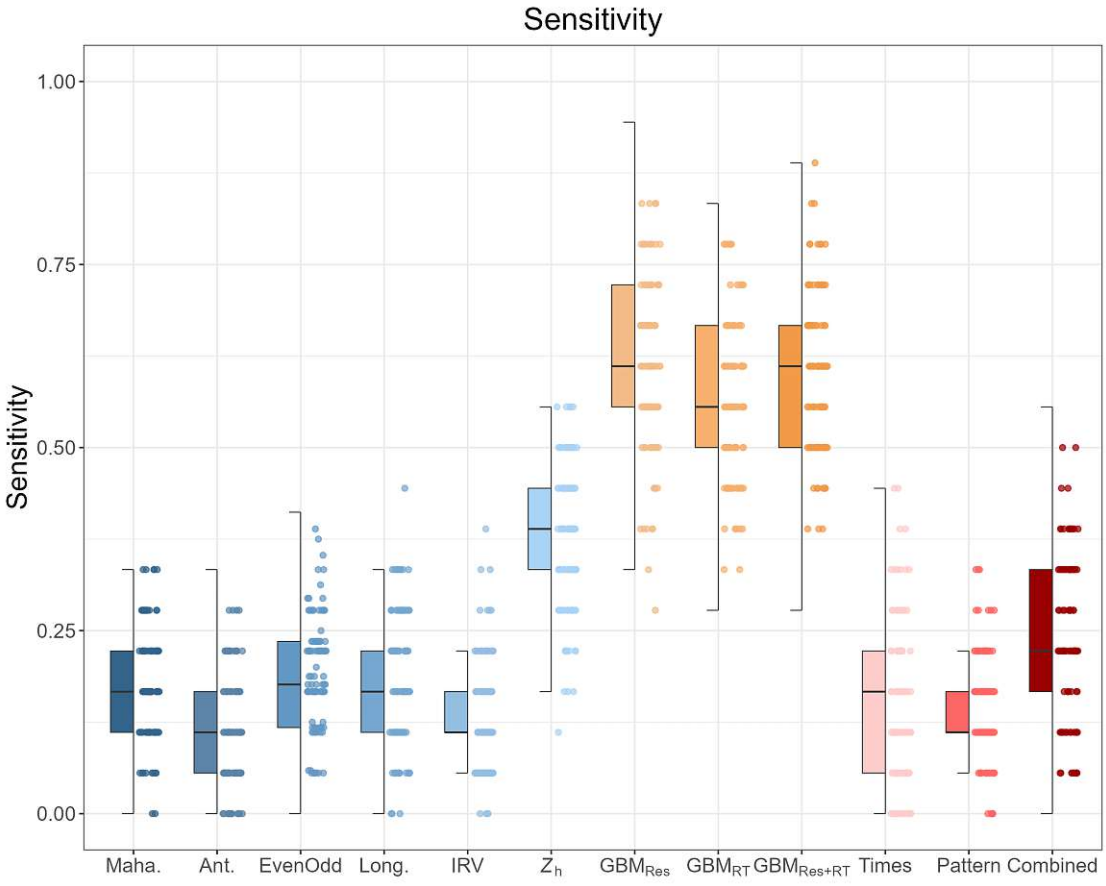

Figure 8.9: Comparison of the sensitivity of different methods using Dataset 2.

It would be possible to increase the sensitivity with our approach if we adapted the thresholds and flagged more respondents. For the answer component, the threshold of 0.4 times the median total answer time is rather conservative and could be increased a bit. We already increased the threshold for the autoencoder to 10% to even have a chance to flag the majority of bad respondents with this component because of the number of bad respondents in the bootstrap sample. This value could also be further increased. However, we want to test our proposed method without significant modifications.

Findings

Even though the sensitivity score is not satisfying, our method outperforms traditional methods while maintaining a higher accuracy score than the methods with better sensitivity scores. The question remains how accurate the labels and therefore how meaningful the results are.

8.3 Effects of not excluding bad interviews

One research question focusses on the effect of not excluding bad respondents. To measure this effect, we create bootstrap samples for both datasets. We create 1000 bootstrap samples by drawing only from the "good" respondents, and 1000 bootstrap samples by drawing from the entire dataset, including the "bad" respondents. For every bootstrap sample, the aggregated survey results are calculated. For Dataset 2, this involves counting how many times each response option is chosen within the sample and then normalizing these counts to get a percentage distribution for every question-answer pair. For Dataset 1, we want representative results of Austria. Therefore, we calculate weights based on demographic features like gender, age and place of living. If a group is underrepresented in the data, the people in the group get higher weights, and if a group is overrepresented, it is weighted down. Then, the weights are used to get the percentages for a selected question.

To determine whether the inclusion of bad respondents significantly alters the survey results, we compare the differences between paired bootstrap samples. Specifically, we calculate the difference between the first sample of the good respondents and the first sample of the full dataset, the difference between the second samples and so on, resulting in 1000 independent observations of differences. This method allows us to estimate the difference between the two distributions directly with the mean which can be seen as the average effect of not excluding bad respondents.

For each question and response option, we check if the response frequency from the full dataset is greater than that from the good respondents. This binary classification can be seen as a Bernoulli-distributed random variable (see “Bernoulli Distribution” (2008)). We calculate the percentage of times the value from the full dataset is greater than the value from only the good respondents. This percentage serves as an estimate of the p of the Bernoulli distribution and indicates how likely it is that the answer percentage of a question-answer pair increases or decreases when bad respondents are excluded.

There are also options to test with a Binomial test (see “Binomial Test” (2008)) if p is equal to 50%, which would indicate no effect. We conduct such tests. Given that multiple comparisons are made across different questions and response options, we apply the Bonferroni correction (see Armstrong (2014)) to control for the increased risk of Type I errors. This correction adjusts the significance threshold, ensuring that our findings of significant differences are not due to random chance.

However, the main focus is not on the testing but on the absolute differences, as they allow for more comprehensive conclusions.

8.3.1 Dataset 1 (Health Survey)

We select one question to analyze the effect of bad respondents. The dataset includes a question which is also used for the calculation of the Healthy Life Years (HLY). Another term that can be used interchangeably for HLY is disability-free life expectancy. The

question to estimate the HLY is called General Activity Limitation Indicator (GALI). For more information see Bogaert et al. (2018) or European Commission (2010). Like the whole survey, the question is in German in the questionnaire, but it translates to English as "How much have you been limited in activities of daily living due to a health problem for at least six months? Would you say you are:" with the answer options "severely limited", "somewhat limited", "not limited" and "no answer".

The effect obviously depends on the number of bad respondents, which is only 22 out of 1028 for Dataset 1. Therefore, a huge shift of the answer percentages is impossible even if bad respondents are not excluded. The low number of bad respondents is because of the closed survey approach and regular checking of the people who get invited to surveys. However, it still seems to make a difference as we see in Figure 8.10. Especially, the result percentage for "no answer" is higher in almost 80% of the bootstrap comparisons if bad respondents are not excluded. If bad respondents had no effect, we would expect a value of around 50%. All answers, except "somewhat limited," are statistically significantly different from 50% ($\alpha = 0.05$) using a Binomial test and Bonferroni correction.

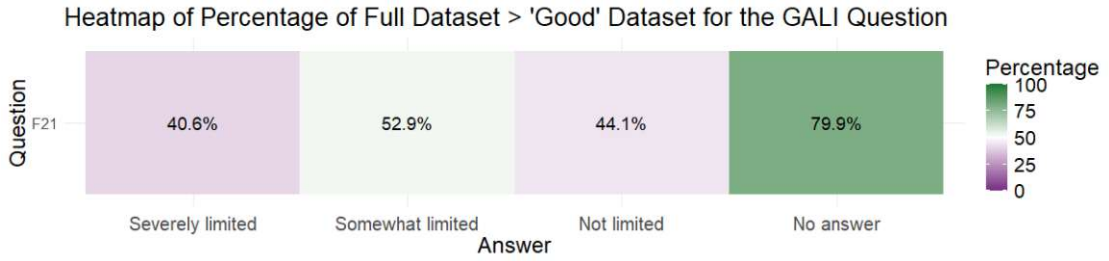

Figure 8.10: Percentage of Full Dataset > 'Good' Dataset for 1000 bootstrap comparisons for the different answers to the GALI-question.

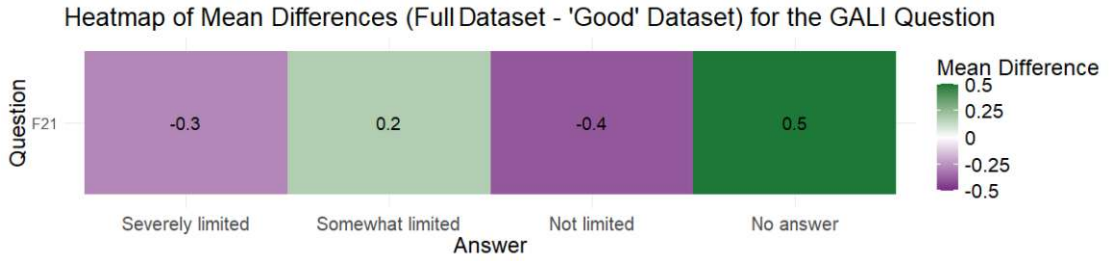

Figure 8.11: Mean of 1000 bootstrap comparisons (Full Dataset - 'Good' Dataset) for the different answers to the GALI-question.

This indicates that there is an effect. The question remains: how big is this effect? Therefore, we calculate all 1000 differences in the answer percentages and take the mean of those. This can be seen as an estimator for the effect size if bad respondents are not excluded from the different answer options. Figure 8.11 shows the mean shifts of the answer percentages for the GALI question. There are average increases of 0.5% and 0.2%

for "no answer" and "somewhat limited", and decreases of 0.4% and 0.3% for "severely limited" and "not limited", respectively. The more extreme answers are less present when the data contains bad respondents. This can lead to an underestimation of those answer options.

If we look at the individual bootstrap distributions without calculating the differences, the distributions look very similar to each other, as we see in Figure 8.12. We see the increase for "no answer", but for all other questions, the differences are only marginal.

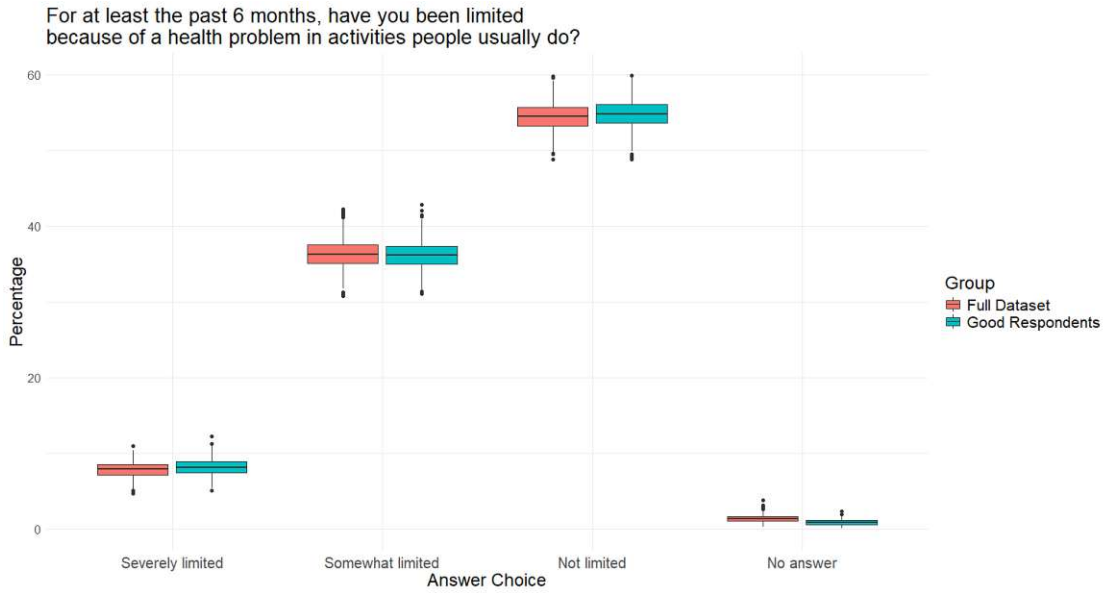

Figure 8.12: Boxplots of all different distributions for the answers to the GALI-question.

This is only one example, but the positive effect size for the answer "no answer" is present throughout the survey. Bad respondents are less interested in answering the survey questions, which is often reflected by refusing to give an answer. Consequently, all other questions are also more likely to contain inaccurate answers because this is an indication of low motivation and carelessness.

The higher percentage for "no answer" leads to lower percentages and an underestimation of other answer options, which in this case are the extreme answer choices. To obtain reliable survey results and avoid such over- or underestimations, bad respondents should be detected and excluded.

8.3.2 Dataset 2 (Literature Survey)

For Dataset 2, there are statements with five possible answers (strong disagreement, disagreement, neutral, agreement, strong agreement). There are 60 statements and Figure 8.13 shows the percentages of how often the answer percentage of the full distribution is higher than that of the good respondents for every question-answer pair. The plot

indicates that only very few values are around 50%. The Binomial test with Bonferroni correction also classifies the majority of question-answer pairs as statistically significantly different ($\alpha = 0.05$) from 50%. For some questions, the "neutral" answer for the full dataset is bigger for almost all comparisons than for the good respondents. The other extreme is "strong agreement," where the value is often close to 0.

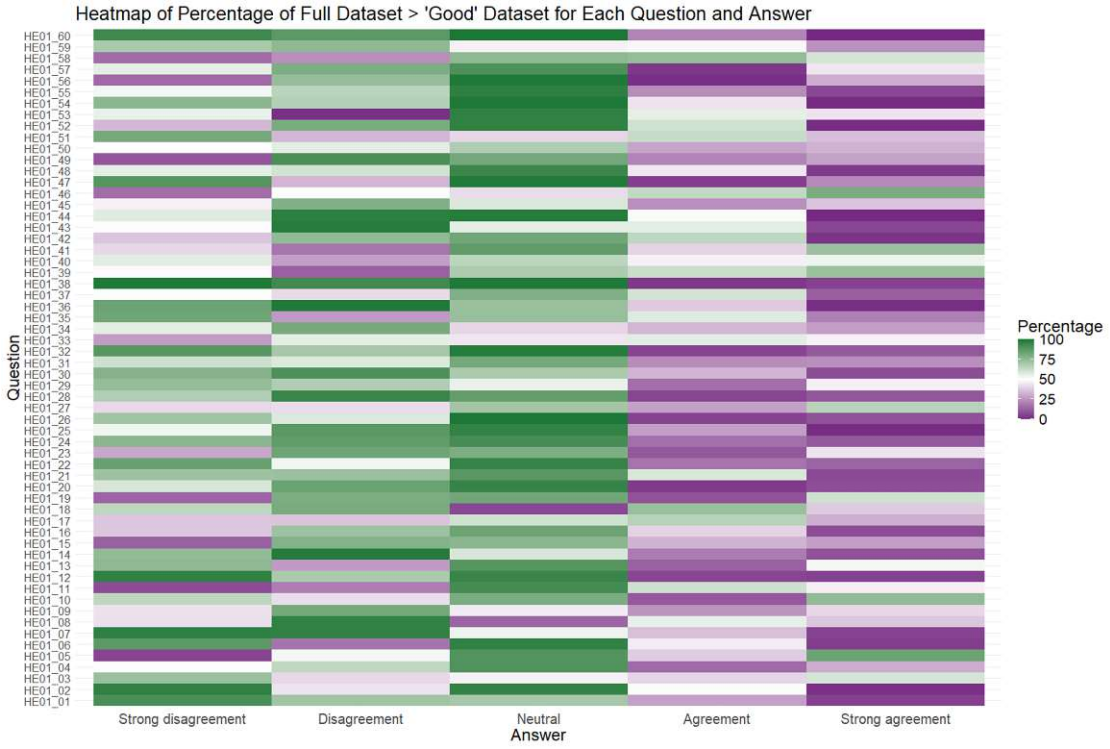

Figure 8.13: Percentage of Full Dataset > 'Good' Dataset for 1000 bootstrap comparisons for every question-answer pair of Dataset 2.

For Dataset 2, 244 out of 605 interviews have a label for exclusion. Therefore, it is not surprising that there is a bigger difference between the full dataset and the good respondents than for Dataset 1. Figure 8.14 shows a heatmap of the mean differences for all comparisons of every question-answer pair. We see that bad respondents are more likely to select one of the first three answer options. Especially the "neutral" option is chosen more frequently by bad respondents. This leads to a strong positive effect size for this answer option if bad respondents are not excluded and a reduction for mainly "strong agreement", which again is an extreme answer option. It might be faster to click on one of the first three options, which might also explain the negative effect sizes for the last two answers. There are shifts in the answer percentages of up to over 5%. Of course, the number of bad respondents is very high in this example, but this gives an idea how much influence bad respondents can have on survey results, especially if there are a lot of bad respondents and nothing is done against them.

Figure 8.14: Mean of 1000 bootstrap comparisons (Full Dataset - 'Good' Dataset) for every question-answer pair of Dataset 2.

There is no option to refuse the answering of a question, but the "neutral" answer comes the closest to "no answer". This aligns with the findings for Dataset 1 that bad respondents are more likely to give this kind of answers.

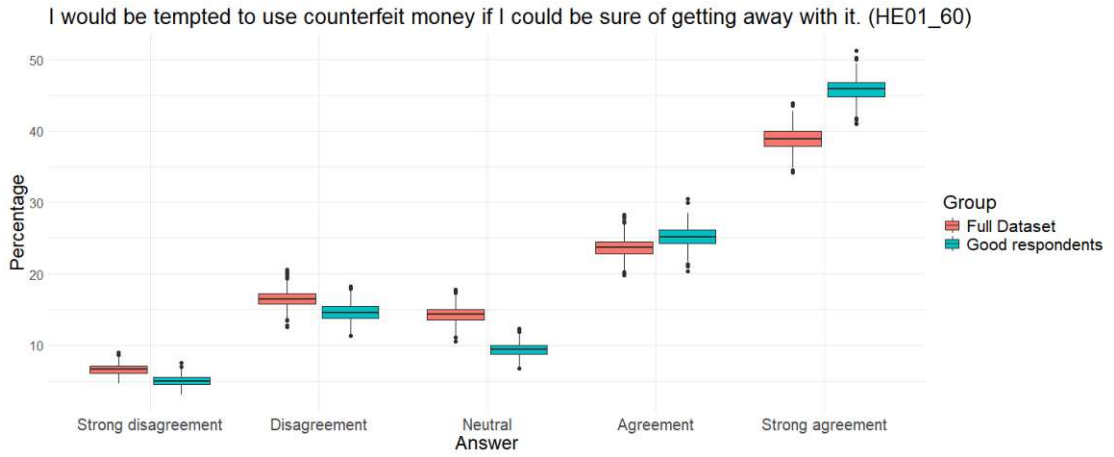

Figure 8.15: Boxplots of all different distributions for the answers to statement "HE01_60".

We select one statement and look at the boxplots of the two distributions. Statement "HE01_60" is "I would be tempted to use counterfeit money if I could be sure of getting away with it.". Figure 8.15 shows the huge differences between the answer percentages of the two distributions. As already mentioned, the "neutral" answer option is more likely

to get chosen by bad respondents. Additionally, it seems that bad respondents shift the answer options to more equal answer percentages. Random responding would lead to a answer distribution of 20% for each answer option. This indicates that alongside the refusing of answering the question, some bad respondents might just respond randomly, which is another characteristic of bad responding.

CHAPTER 9

Conclusion

We propose a new method to detect useful interviews in closed online surveys. It uses cluster algorithms to find clusters of people responding too fast on a question level and the reconstruction error of an autoencoder to detect anomalies in the answer patterns. Compared to traditional methods, we are able to find a higher proportion of bad respondents while maintaining similar or even better overall accuracies. The two components of our strategy are answer times and answer patterns. One finding is that both are important indicators of bad respondents and both should be used because the two different components detect different kinds of bad responding. Our experiments show that if an interview is flagged by the time component, one can quite confidently eliminate the interview as it is almost surely the right decision. We also see that the analysis on a question level should be preferred over using the total response times. For the answer pattern component, one has to be careful to not blindly eliminate all flagged interviews. Sometimes also reasonable interviews with different or extreme answer behaviours are flagged by the autoencoder. However, we find that bad interviews are more likely to have higher reconstruction errors. Respondents choosing very often "no answer" or providing inconsistent answers are flagged by the autoencoder. Therefore, an autoencoder is capable of detecting bad respondents. We recommend to further investigate the interviews flagged by the autoencoder and finally decide if the interview should be excluded or not.

The analysis of the influence of bad respondents on the aggregated survey results shows that it can lead to statistically significant changes in the answer percentages if bad respondents are not excluded, which highlights the importance of the topic. The willingness of answering questions is decreased for bad respondents as they do not answer questions or give neutral answers more often. People refusing excessively to answer questions should be excluded since the answers to other questions cannot be trusted because of the careless answer behaviour.

Overall, the difficulty of the task lies in the fact that one can never be sure if an interview is rightfully included or excluded. The reason is that there is never a guarantee whether

the answers provided are true or not. It is very important to be aware of the possibility of bad respondents and take measures to get rid of them as they might falsify the results of the survey. Ideally, the initial number of bad respondents is low. This can be achieved with preventive measures such as using closed surveys as the Austrian company OGM research & communication GmbH does. If a closed survey is not possible, one can include questions with attention checks to make it easier to filter out bad respondents afterwards. Especially if no further information is available, our strategy can be used to preselect useful interviews. Additional survey specific checks can and should be done if possible.

Further research could include experiments if this method is also suitable for bot detection. As with almost every neural network, the decision of the autoencoder used can be hardly traced. There are efforts in the literature to make autoencoders more explainable. It might also be interesting to include this explainability into our strategy to see better which questions lead to high reconstruction errors.

Overview of Generative AI Tools Used

The only generative AI-tool I used for this thesis was ChatGPT.

The coding part was written by me, with assistance from ChatGPT for debugging and generating a few code snippets. I generated code snippets for libraries and commands I was not so familiar with and used them as a starting point for my own coding.

This diploma thesis was proofread with the assistance of ChatGPT. I used it to translate words or phrases when I couldn't find the right English term, and to help express ideas more clearly when I was unsure how to phrase them. Additionally, I relied on it to refine formulations that I felt could be improved. The ideas and arguments of this thesis were not generated by ChatGPT.

List of Figures

List of Tables

List of Algorithms

Bibliography

Alvarez, R. M., & Li, Y. (2023). Survey Attention and Self-Reported Political Behavior. *Public Opinion Quarterly*, *86*(4), 793–811.

Armstrong, R. A. (2014). When to use the Bonferroni correction. *Ophthalmic and Physiological Optics*, *34*(5), 502–508.

Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, 353–374.

Belete, D., & Manjaiah, D. H. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, *44*, 1–12.

Bernoulli Distribution. (2008). In *The Concise Encyclopedia of Statistics* (pp. 36–37). Springer New York.

Bhattacherjee, A. (2012). *Social Science Research: Principles, Methods, and Practices*. Textbooks Collection.

Binomial Test. (2008). In *The Concise Encyclopedia of Statistics* (pp. 47–49). Springer New York.

Bogaert, P., Van Oyen, H., Beluche, I., Cambois, E., & Robine, J.-M. (2018). The use of the global activity limitation Indicator and healthy life years by member states and the European Commission. *Archives of Public Health*, *76*, 30.

Bonett, S., Lin, W., Sexton Topper, P., Wolfe, J., Golinkoff, J., Deshpande, A., Villarruel, A., & Bauermeister, J. (2024). Assessing and Improving Data Integrity in Web-Based Surveys: Comparison of Fraud Detection Systems in a COVID-19 Study. *JMIR Form Res*, *8*.

Buskirk, T. D., Kirchner, A., Eck, A., & Signorino, C. S. (2018). An Introduction to Machine Learning Methods for Survey Researchers. *Survey Practice*, *11*(1).

Chen, S., & Guo, W. (2023). Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics*, *11*(8), 1–54.

Curran, P. G. (2016). Methods for the detection of carelessly invalid responses in survey data. *Journal of Experimental Social Psychology*, *66*, 4–19.

Daikeler, J., Bosnjak, M., & Manfreda, K. (2020). Web Versus Other Survey Modes: An Updated and Extended Meta-Analysis Comparing Response Rates. *Journal of Survey Statistics and Methodology*, *8*, 513–539.

Drasgow, F., Levine, M. V., & Williams, E. A. (1985). Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology*, *38*(1), 67–86.

Dunn, A. M., Heggestad, E. D., Shanock, L. R., & Theilgard, N. (2018). Intra-individual Response Variability as an Indicator of Insufficient Effort Responding: Comparison to Other Indicators and Relationships with Individual Differences. *Journal of Business and Psychology*, *33*(1), 105.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.

European Commission. (2010). Healthy Life Years (HLY) - Page Updated [Accessed: 2024-08-17]. https://health.ec.europa.eu/latest-updates/healthy-life-years-hly-page-updated-2010-06-30_en

Felt, J., Castaneda, R., Tiemensma, J., & Depaoli, S. (2017). Using Person Fit Statistics to Detect Outliers in Survey Research. *Frontiers in Psychology*, *8*, 863.

Fradkov, A. L. (2020). Early History of Machine Learning. *IFAC-PapersOnLine*, *53*(2), 1385–1390.

Fricker Jr, R. D. (2016). Sampling Methods for Online Surveys. *The SAGE Handbook of Online Research Methods*, 162–183.

Gaur, P., Zimba, O., Agarwal, V., & Gupta, L. (2020). Reporting Survey Based Studies - a Primer for Authors. *Journal of Korean Medical Science*, *35*(45), e398.

Glazer, J., MacDonnell, K., Frederick, C., Ingersoll, K., & Ritterband, L. (2021). Liar! Liar! Identifying eligibility fraud by applicants in digital health research. *Internet Interventions*, *25*.

Gomes, C., Jin, Z., & Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *Journal of Risk and Insurance*, *88*(3), 591–624.

Goodrich, B., Fenton, M., Penn, J., Bovay, J., & Mountain, T. (2023). Battling bots: Experiences and strategies to mitigate fraudulent responses in online surveys. *Applied Economic Perspectives and Policy*, *45*(2), 762–784.

Greene, J., Speizer, H., & Wiitala, W. (2008). Telephone and Web: Mixed-Mode Challenge. *Health services research*, *43*, 230–48.

Greszki, R., Meyer, M., & Schoen, H. (2015). Exploring the Effects of Removing "Too Fast" Responses and Respondents from Web Surveys. *Public Opinion Quarterly*, *79*, 471–503.

Griffin, M., Martino, R. J., LoSchiavo, C., Comer-Carruthers, C., Krause, K. D., Stults, C. B., & Halkitis, P. N. (2022). Ensuring survey research data integrity in the era of internet bots. *Quality and Quantity*, *56*(4), 2841–2852.

Groves, R. M. (2011). Three eras of survey research. *The Public Opinion Quarterly*, *75*(5), 861–871.

Hartigan, J. A. (1975). *Clustering Algorithms* (99th). John Wiley & Sons, Inc.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Jadhav, A., Pramod, D., & Ramanathan, K. (2019). Comparison of Performance of Data Imputation Methods for Numeric Dataset. *Applied Artificial Intelligence*, *33*, 1–21.

Jebreel, N. M., Haffar, R., Singh, A. K., Sánchez, D., Domingo-Ferrer, J., & Blanco-Justicia, A. (2020). Detecting Bad Answers in Survey Data Through Unsupervised Machine Learning. In J. Domingo-Ferrer & K. Muralidhar (Eds.), *Privacy in Statistical Databases* (pp. 309–320). Springer International Publishing.

Johnson, J. A. (2005). Ascertaining the validity of individual protocols from Web-based personality inventories [Proceedings of the Association for Research in Personality]. *Journal of Research in Personality*, *39*(1), 103–129.

Kelley, K. (2003). Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, *15*, 261–266.

Kennedy, C., Hatley, N., Lau, A., Mercer, A., Keeter, S., Ferno, J., & Asare-Marfo, D. (2021). Strategies for Detecting Insincere Respondents in Online Polling. *Public Opinion Quarterly*, *85*(4), 1050–1075.

Kieu, T., Yang, B., Guo, C., Jensen, C. S., Zhao, Y., Huang, F., & Zheng, K. (2022). Robust and explainable autoencoders for unsupervised time series outlier detection. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3038–3050.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, 1137–1143.

Lipps, O., Herzing, J. M. E., Pekari, N., Stähli, M. E., Pollien, A., Riedo, G., & Reveilhac, M. (2019). *Incentives in Surveys* (FORS Guide No. 08). Swiss Centre of Expertise in the Social Sciences FORS. Lausanne.

Mahalanobis, P. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science of India*, *2*, 49–55.

Manfreda, K. L., Bosnjak, M., Berzelak, J., Haas, I., & Vehovar, V. (2008). Web Surveys versus other Survey Modes: A Meta-Analysis Comparing Response Rates. *International Journal of Market Research*, *50*(1), 79–104.

Meade, A., & Craig, B. (2012). Identifying Careless Responses in Survey Data. *Psychological methods*, *17*, 437–55.

Nazeer, K. A., & Sebastian, M. (2009). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. *Proceedings of the World Congress on Engineering*, *1*, 1–3.

Nordholt, E. S. (1998). Imputation: Methods, Simulation Experiments and Practical Examples. *International Statistical Review*, *66*(2), 157–180.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Courna-

peau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pinzón, N., Koundinya, V., Galt, R., Dowling, W., Boukloh, M., Taku-Forchu, N. C., Schohr, T., Roche, L., Ikendi, S., & Cooper, M. H. (2023). *AI-Powered Fraud and the Erosion of Online Survey Integrity: An Analysis of 31 Fraud Detection Strategies* (tech. rep.). Center for Open Science.

Pratesi, M., & Salvati, N. (2006). Missing Data and Small-Area Estimation: Modern Analytical Equipment for the Survey Statistician. Nicholas T. Longford. *Journal of the American Statistical Association*, *101*, 1729–1730.

Read, B., Wolters, L., & Berinsky, A. (2021). Racing the Clock: Using Response Time as a Proxy for Attentiveness on Self-Administered Surveys. *Political Analysis*, *30*, 1–20.

Roopa, S., & Menta Satya, R. (2012). Questionnaire Designing for a Survey. *The Journal of Indian Orthodontic Society*, *46*, 37–41.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, *65*, 386–408.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Rubin, D. B., & Schenker, N. (1991). Multiple imputation in health-are databases: An overview and some applications. *Statistics in Medicine*, *10*(4), 585–598.

Rumelhart, D. E., & McClelland, J. L. (1987). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (pp. 318–362).

Sabetti, L., & Heijmans, R. (2021). Shallow or deep? Training an autoencoder to detect anomalous flows in a retail payment system. *Latin American Journal of Central Banking*, *2*(2), 1–14.

Sakshaug, J. W., Beste, J., & Trappmann, M. (2023). Effects of mixing modes on nonresponse and measurement error in an economic panel survey. *Journal for Labour Market Research*, *57*(1), 1–16.

Särndal, C., Swensson, B., & Wretman, J. (2003). *Model Assisted Survey Sampling*. Springer New York.

Schroeders, U., Schmidt, C., & Gnambs, T. (2022). Detecting Careless Responding in Survey Data Using Stochastic Gradient Boosting. *Educational and Psychological Measurement*, *82*(1), 29–56.

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 464–472.

Smith, T., & Kim, J. (2015). A Review of Survey Data-Collection Modes: With a Focus on Computerizations. *Riron to Hoho / Sociological Theory and Methods*, *30*, 185–200.

Soland, J. G., Kuhfeld, M., & Rios, J. A. (2021). Comparing different response time threshold setting methods to detect low effort on a large-scale assessment. *Large-scale Assessments in Education*, *9*, 1–21.

Statistik Austria. (2024). ICT Usage in Households [Accessed: 2024-08-17]. https://www.statistik.at/statistiken/forschung-innovation-digitalisierung/digitale-wirtschaft-und-gesellschaft/ikt-einsatz-in-haushalten

Storozuk, A., Ashley, M., Delage, V., & Maloney, E. (2020). Got Bots? Practical Recommendations to Protect Online Survey Data from Bot Attacks. *The Quantitative Methods for Psychology*, *16*, 472–481.

Ternovski, J., Orr, L., Kalla, J., & Aronow, P. (2022). A Note on Increases in Inattentive Online Survey-Takers Since 2020. *Journal of Quantitative Description: Digital Media*, *2*, 1–35.

Ulitzsch, E., Pohl, S., Khorramdel, L., Kroehne, U., & von Davier, M. (2024). Using Response Times for Joint Modeling of Careless Responding and Attentive Response Styles. *Journal of Educational and Behavioral Statistics*, *49*(2), 173–206.

Ulitzsch, E., Yildirim-Erbasli, S. N., Gorgun, G., & Bulut, O. (2022). An explanatory mixture IRT model for careless and insufficient effort responding. *British Journal of Mathematical and Statistical Psychology*, *75*(3), 668–698.

Varaine, S. (2023). How Dropping Subjects Who Failed Manipulation Checks Can Bias Your Results: An Illustrative Case. *Journal of Experimental Political Science*, *10*(2), 299–305.

Voorpostel, M., Lipps, O., & Roberts, C. (2021). Mixing Modes in Household Panel Surveys: Recent Developments and New Findings. *Advances in Longitudinal Survey Methodology* (pp. 204–226). John Wiley & Sons, Ltd.

Wardropper, C., Dayer, A., Goebel, M., & Martin, V. (2021). Conducting conservation social science surveys online. *Conservation biology: the journal of the Society for Conservation Biology*, *35*(5), 1650–1658.

World Association for Public Opinion Research. (2014). ESOMAR/WAPOR Guideline on Opinion Polls and Published Surveys [Accessed: 2024-08-17]. https://wapor.org/wp-content/uploads/esomar-wapor-guideline-on-opinion-polls-and-published-surveys-english-august-2014.pdf

Yentes, R., & Wilhelm, F. (2023). *careless: Procedures for computing indices of careless responding* [R package version 1.2.2].

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *SIGMOD Rec.*, *25*(2), 103–114.