Anomaly Detection in Online Search Behavior of Potential Customers within Paid Search Advertisements

by

Tomas Liesting

Master Thesis
Student ID: 453177
Thesis supervisor: dr. Andreas Alfons
Co-reader: dr. Mikhail Zhelonkin

Master of Science in Econometrics and Operations Research
Erasmus University Rotterdam
Business Analytics and Quantitative Marketing
February 2023

Abstract

Within paid search, advertisers compete for top listing positions in search engines based on the user's query. Monitoring impressions, clicks, return on ad spend (ROAS), and other variables in the digital marketing funnel within different markets or paid-search tactics can be tedious, especially when one wants to do this regularly. This research aims to automatically detect anomalies in this search and click behavior of potential customers in the aviation industry through an assignment of the KLM, through which the data was collected. The created method extends Detect Deviating Cells (DDC) (Rousseeuw & Bossche, 2018), a method to detect cell-wise outliers in a two-dimensional dataset, by extending it with a robust outlier detection method in univariate time series and a trend heuristic. The model is evaluated by experts of KLM and with a simulation study. Experts were satisfied as some previously undiscovered outliers were found in the data. The simulation method performed best on 10% contaminated data with large outliers and yielded an F1-score of 0.87.

Contents

Chapter 1

Introduction

In digital marketing, data-driven methods are becoming increasingly important (Braverman, 2015). Different key performance indicators (KPIs) are often used to monitor, for example, the performance of products, markets, or campaigns, and marketing strategies are adjusted accordingly (Ghahremani-Nahr & Nozari, 2021). Especially in paid search, also known as sponsored search, an advertising method with which advertisers compete for top listing positions in search engines based on the user's query (Laffey, 2007). Due to the extensive usage of search engines, paid search advertising has grown to a multi-billion dollar marketing channel (Rutz & Bucklin, 2011). Amongst others, the aviation industry uses paid search often (Burger et al., 2013). E.g., when searching for 'tickets to New York', many advertisements pop up depending on your location and search history. These paid search advertisements generally aim to nudge search-engine users looking for a specific item to the respective company's website, such that these users will eventually 'convert' (purchase on that website). These advertisements can also be combined with different campaigns (like discounts), varying across markets.

The ranking of these results is based on a complex, and for most search engines not wholly revealed, algorithm, which depends on the quality score and the bid price (Laffey, 2007). The quality score depends on the similarity between the search query and the website. This means that an online shop that sells mountain bikes will have a high quality score when the search query is 'buy mountain bikes,´ while an online store selling clothes will have a low quality score. The bid price is the willingness to pay for a click and is set by the company that places the ad, reflecting the valuation of a click. The ad rank will eventually be the quality score multiplied by the bid price. Furthermore, the number of keywords relevant to a company can be enormous. For an airline, one can imagine that 'flights from Amsterdam to New York' can be just as relevant as 'tickets to South Africa', while there are almost no overlapping terms.

The search behavior of users looking at the advertisements can currently be tracked. More specifically, this means that a company can see how many times specific ads are seen, how many times a user clicks on its advertisement, and whether they put something in their cart or bought something. This data is aggregated in a time series, for example, the number of clicks per day. For a company, this data can give insights into emerging trends or the increasing popularity of specific products, as well as into conversion rates and effective bidding strategies. This data also gives insight into keywords that work well and those that do not. For example, when the US plays a football match against the Netherlands, and many people are searching for 'tickets US-Netherlands' with the intention of buying a ticket to a football match, the number of searches for this query will peak, but an airline company would not want to spend much money to be the highest ranked ad in the search results, as consumers are probably not looking for plane tickets. This implies that the number of times the ad is seen will be high when bidding on this query specifically, but the conversions from this advertisement will likely be low.

Additionally, as the bidding generally is based on an automatic process, the amount spent and willingness to pay is often determined by an algorithm, meaning that the amount spent increases when more potential customers enter relevant search terms and vice versa. However, this could result in a spiral, meaning there are fewer conversions because of less spending leading to fewer conversions, or the other way around. One can imagine that a company wants to detect if odd things are happening, called outlier detection or anomaly detection, to monitor KPIs, data quality, and costs. Detecting these anomalies by hand can be time-consuming, as there can be many different combinations of search queries, countries, and campaigns, and anomalies can happen for different KPIs, which is why an algorithm being able to detect these different anomalies would be desirable.

In this research, the dataset of KLM is used to attempt to detect these kinds of anomalies in paid search advertising, leading to the research question:

How can we detect anomalies in the search- and click behavior of potential customers in the aviation industry?

To the best of my knowledge, no other research has yet investigated anomalies in paid search advertisements for airlines specifically. Airlines have an additional dimension of depending on the type of flight a potential customer is looking for. KLM only wants to advertise flights that they fly themselves, and therefore has to employ different tactics in different countries. To

detect these kinds of anomalies, the granularity of the data is very important. For example, one would like to be able to detect anomalies in search behavior for tickets from Moscow to Warsaw. However, if only ten searches per month are done for this flight, and one group buys ten tickets because they are going with their sports team, this should not be qualified as an anomaly, even though it is a large increase. Furthermore, we cannot spot these anomalies at more specific levels if the aggregation level is too high. One also has to take into account the business side of the model. A large airline company would not be bothered by ten tickets more or less on a specific day, as it sells tickets worth millions of euros on a daily basis. This, combined with the fact that the outliers should also be actionable, means that they do not want to waste their time investigating outliers that they cannot do anything about. Additionally, for paid search in general, to the best of our knowledge, only simple statistical outlier-detection techniques have been applied (Peck, 2011). These techniques are Tukey's 1.5 inter-quartile range (IQR) method, the two median absolute deviation method, and the three standard deviations method. In this research, I will investigate different outlier detection techniques, after which one method will be selected.

In order to answer the research question, three logical sub-questions follow. First, what do we mean by search- and click behavior in this context? Second, how to define anomalies in this context? And third, which anomaly detection algorithms are relevant to detect these anomalies in this context?

To start with the first question, a consumer's search and click behavior starts from the moment a consumer searches for a relevant keyword until the consumer buys a product. This is called the marketing funnel, for which KLM has a specific setup. This starts with creating awareness and building your brand, which is achieved by broad communication, without any specific targeting. Examples of these are TV commercials, organizing public events, etc. The next steps are brand consideration, product consideration, and booking intent. This starts when a potential customer is interested in an airplane ticket and starts searching. When someone searches 'airplane tickets,' one would want this user to click on your airline website and plan her journey here. More specifically, when someone goes to the ticket booking page of the website, this is called booking intent and is the next step in the funnel phase. The next step is when the customer buys something from the website and books a ticket, called (tactical) sales. Next, the company attempts to upsell, which means offering extra luggage, extra leg space, etc. The last steps are loyalty and service, which means that a customer remains a customer for a long time. In this research, I want to detect anomalies from the brand consideration step until upselling.

This means that the customer journey starts when they realize they want to buy an airplane ticket and start looking for options, and ends when the booking is made and extra options are bought. In Section 3, I will elaborate on which data is available on this funnel.

For the second and third questions, as outlier detection is a large subject in the academic literature, Chapter 2 will first give an overview of the types of outliers, after which I will specify what kind of outliers are relevant in this research. Additionally, I consider different anomaly detection techniques based on the literature and determine which algorithm is relevant for this research.

Chapter 2

Literature

Anomaly detection is a broad term encompassing many different techniques. It also has different names, such as novelty detection, outlier detection, exception mining, or noise detection (Hodge & Austin, 2004). In this paper, we will use the term anomaly or outlier detection, as I judge this as the most appropriate term for the task. Anomaly detection can also have multiple definitions. One widely used comes from Hawkins (1980), who defines an outlier as "An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism." Outliers are, therefore, observations that do not follow expected behavior, whereas unexpected behavior can differ from situation to situation. The detection method depends on the type of outlier and the type of input data. Due to the many different applications of anomaly detection methods, many different categorizations have been made. As I first must clarify what kind of anomaly detection problem we have, I will cover different aspects of anomaly detection problems. Afterward, I will select the relevant specification of the problem and will cover appropriate techniques.

2.1 Aspects of outlier detection

There are many different categorizations of outlier detection techniques (Munir et al., 2019; Aggarwal, 2017; Hodge & Austin, 2004; Blázquez-García et al., 2021; Chandola et al., 2009). I consider four different aspects of anomaly detection.

First, the literature considers the type of input data (Blázquez-García et al., 2021; Chandola et al., 2009). They describe input data as a collection of data instances containing attributes like variables, dimensions, or fields. These attributes can be binary, categorical, or continuous. If a data instance consists of only one attribute, we call the input data univariate. We call the input

data multivariate if a data instance consists of more than one attribute. Also, the data instances can be related to each other. For example, data instances can be sequence, spatial, or graph data. In sequence data, the observations are ordered, like time series data, while spatial data means that data instances are related to instances close by. In graph data, the data instances are connected with edges, and data instances are vertices.

Secondly, the nature of the anomaly should be considered (Munir et al., 2019; Chandola et al., 2009; Hodge & Austin, 2004; Blázquez-García et al., 2021; Goldstein & Uchida, 2016). Outliers can be individual or collective/subsequent, where individual outliers imply that only a single data instance is outlying, and collective outliers are a group of anomalies. Blázquez-García et al. (2021) also consider time series anomalies, only relevant for multivariate time series, where a whole time series is considered anomalous compared to the other variables when for example, one time series has a decreasing trend while all other time series are increasing. Furthermore, a distinction is made between global and local anomalies (Blázquez-García et al., 2021; Goldstein & Uchida, 2016), or point and contextual anomalies (Chandola et al., 2009; Munir et al., 2019). Both refer to whether observations are irregular by themselves or whether the data instances are only outlying within a specific context, such as a specific time frame (Blázquez-García et al., 2021) or a subset of a graph (Goldstein & Uchida, 2016). The distinction between the two specifications is that papers referring to local and global anomalies generally consider non-sequence data, while point and contextual anomalies do (Blázquez-García et al., 2021).

Thirdly, categorization is based on the level of supervision. For outlier detection, this implies the presence of data labels on whether a data instance is considered normal or anomalous (Peck, 2011). Based on the presence of these labels, one can select one of three different methods for detecting anomalies, namely supervised, semi-supervised, or unsupervised (Hodge & Austin, 2004; Chandola et al., 2009; Blázquez-García et al., 2021; Goldstein & Uchida, 2016; Aggarwal, 2017). In supervised anomaly detection, the data consists of fully labeled train and test sets on which standard classifiers can be trained, where generally classes are unbalanced (Goldstein & Uchida, 2016), meaning that the group of outliers is smaller than the group of normal observations. In semi-supervised approaches, only examples of 'normal' data points are available (Aggarwal, 2017). In other words, the normal observations are modeled, and a point is considered an outlier if it deviates too much from what is 'normal' (Chandola et al., 2009). Unsupervised approaches assume that outliers will be separated from the normal observations based on their intrinsic properties (Goldstein & Uchida, 2016).

Fourthly, the output of the anomaly detection technique should be considered (Goldstein &

Uchida, 2016; Chandola et al., 2009; Peck, 2011). Outputs can be scores or labels, where scores give anomaly scores like numeric values to data instances, where an analyst can determine a certain threshold or cutoff point to select the anomalies, and where techniques with labels assign actual labels or classes (normal or abnormal) to different data instances (Chandola et al., 2009).

2.2 Nature of outlier detection techniques

Next to determining aspects of outlier detection, the nature of the method should be considered. One distinction has already been made in the previous section: supervised, semi-supervised, and unsupervised. However, another categorization is based on the nature of the underlying method. Many different classifications are possible, depending on the goal of the author. Also, authors make different choices on which nature of the method falls in which category. Some only distinguish between statistical and neural-network-based methods (Markou & Singh, 2003; Munir et al., 2019), or add a separate machine learning category (Hodge & Austin, 2004). These papers consider clustering- and nearest-neighbor-based approaches to qualify as statistical methods. Other authors consider these approaches separate groups (Zhang et al., 2019; Goldstein & Uchida, 2016; Chandola et al., 2009). Generally, most papers do consider similar methods, although the categorization differs. I choose to cover statistical models, classification-based models, nearest neighbor-based models, and clustering-based approaches, as I believe most anomaly detection methods in the literature are covered with this classification.

Statistical anomaly detection techniques model the data based on statistical properties and determine whether test samples come from a similar distribution (Markou & Singh, 2003). These statistical techniques can be parametric or non-parametric, where hybrid semi-parametric approaches are also possible (Hodge & Austin, 2004). For parametric approaches, the most frequent assumption is some kind of Gaussian distribution (Markou & Singh, 2003). One can determine whether an observation is an outlier if the score lies above a certain threshold (Peck, 2011). Other assumptions of the distributions are possible. Also, regression model-based outlier detection techniques assume distributions of the data. These models are often used in time series and attempt to fit a model on the data, after which the residuals of the fitted values are calculated (Blázquez-García et al., 2021; Munir et al., 2019). These residuals then determine whether a point is outlying or not. The fitting of the model may be affected by the outliers, creating a bias in the model; however, extensive research has been done on the robust fitting of regressions and ARIMA models to make the fitting of the model more robust against outliers (Bianco et

al., 1996; Maronna, 2017). Non-parametric methods do not make assumptions about statistical properties of the data (Markou & Singh, 2003). Examples are Histogram-based methods (Munir et al., 2019; Chandola et al., 2009), where a histogram is created from the given data. If a test data point falls within one of the bins, the point would be considered normal, while the opposite would be true for data points that do not fall in a certain bin. The advantages of statistical techniques are that they are justified if the underlying assumption holds and that the technique can operate without any labeled training data if the distribution estimation is robust to outliers (Markou & Singh, 2003). The obvious disadvantage is that the technique depends on a parametric distribution, which does not always hold, especially in higher dimensions (Chandola et al., 2009). Also, the best test statistics to determine whether a point is an anomaly cannot easily be found, as many are available, particularly in higher dimensions.

Classification-based models assume that the distinction between normal and anomalous cases can be learned, as with supervised or semi-supervised models, but where no labeled data instances are necessary (Zhang et al., 2019). These classification models can be one-class or multi-class, implying that there exists either one normal class or multiple normal classes (Hodge & Austin, 2004). Classification-based approaches follow a similar two-step procedure as in regression-based models, where a model is fitted on the (normal) data, after which outliers are detected. Generally, neural network-based approaches are classification-based approaches, as the neural network generally forms a classifier (Hodge & Austin, 2004). Therefore, Chandola et al. (2009) distinguish between different kinds of classification-based approaches. One is a neural network-based approach, while this is a separate class in other work (Zhang et al., 2019; Munir et al., 2019; Hodge & Austin, 2004; Markou & Singh, 2003). I decided to include neural network-based and machine learning-based approaches in the classification-based models, as they generally overlap. Examples are the work of Munir et al. (2019), who created a deep convolutional neural network that is fitted on time series data, after which anomalies are detected based on tagging periods as normal or abnormal. Also, shallow neural networks (i.e. other machine learning classification algorithms) can be used, such as support vector machines (Hodge & Austin, 2004), but also rules determined by experts can be a way of classifying abnormalities (Chandola et al., 2009). The advantage of these models is that they are powerful if data labels are available, but the largest disadvantage is that with no available 'normal' data instances, these kinds of models cannot be used (Chandola et al., 2009).

Nearest neighbor-based (NN-based) models assume that normal observations occur in dense neighborhoods, while anomalies are generally further away from their closest neighbors (Chan-

dola et al., 2009). To do this, a distance or similarity measure must be defined based on the data, which is also the case for clustering-based approaches. Anomaly detection methods that require a distance metric are sometimes called distance-based outliers (Angiulli & Fassetti, 2007). The Euclidean distance is often used for continuous data, but other distance metrics can be appropriate. For multivariate data, distances are usually calculated between individual variables, after which they are combined. An example of determining an anomaly score is calculating its distance to its nearest neighbors; if this distance is large, a point is considered anomalous (Markou & Singh, 2003). Another possibility is to use the relative density. This means that data points that lie in a dense neighborhood are normal, while low-density points are abnormal (Chandola et al., 2009). To illustrate, consider Figure 2.1, where global nearest neighbor techniques such as the k-nearest neighbor approach might not classify p_2 anomaly due to the low density of cluster C_1, but a local density-based approach will, while p_1 would be considered an outlier in both approaches. One widely used model is the Local Outlier Factor (Breunig et al., 2000). This model uses the ratio between the average local density of the k nearest neighbors and the local density of the data point to create an anomaly score. The advantages of nearest neighbor-based models are that they are unsupervised and, therefore, purely data-driven, and are relatively straightforward to implement, also for different data types. One 'only' needs an appropriate distance measure (Chandola et al., 2009). Disadvantages are that normal observations do not necessarily have to be close to other observations, and abnormal observations are not necessarily far. For example, when comparing markets, a large market with many sales and views is not necessarily anomalous. Also, computational complexity is a challenge, and it relies greatly on the distance measure that is picked. Lastly, nearest neighbor-based approaches do not naturally take the structure of the time series into account, making no clear distinction between points that are observed at the beginning of the sequence or the end of the sequence (Chandola et al., 2009).

Fourthly we consider clustering-based approaches. These approaches rely on the assumption that normal observations belong in a cluster, while anomalies do not (Markou & Singh, 2003). These algorithms also consist of two steps. First, the data points are clustered, and second, it creates anomaly scores for data instances based on the distance from the cluster's center. Examples are the use of DBSCAN, but also Cluster-Based Local Outlier Factor models are used. This method is similar to the NN-based models; however, the NN-based models rely on distances between two data instances, while clustering-based techniques evaluate instances based on the cluster it belongs to (Chandola et al., 2009). Advantages and disadvantages are

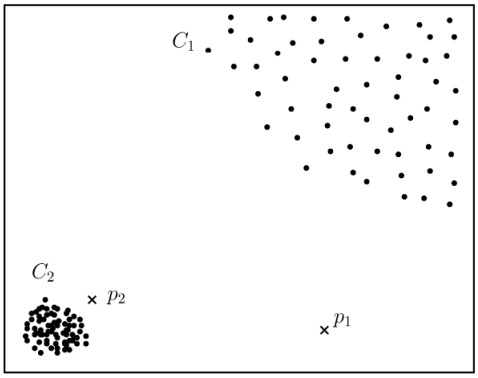


Figure 2.1: Illustrating the advantage of local density-based techniques and general nearest neighbor techniques. Retrieved from Chandola et al. (2009)

relatively overlapping; however, clustering-based techniques additionally assume that normal data instances are relatively close to the centroid of the cluster, which does not necessarily have to be the case (Markou & Singh, 2003).

2.3 Outlier detection in paid search advertisements

The problem of this research concerns paid search advertisements, as described earlier. For outlier detection in paid search specifically, not a lot of public research has been done. Peck (2011) investigated how to quickly find potential outliers in a univariate case by using techniques such as a standard deviation method, where any number higher than two or three times the standard deviation is considered an outlier, Tukey's method, which uses an interquartile range of either 1.5 or 3, and a median absolute deviation (MAD) method, which uses two to three times the MAD as a range to detect outliers. Though these methods are handy for finding univariate global outliers, the relation between the variables as well as the temporal relationships is not taken into account.

2.4 Cell-Wise Outlier Detection

For our research, our input data is multivariate, with discrete or continuous values. For example, the number of clicks or the number of bookings can be considered discrete, as half-bookings are not allowed. However, we do consider all of our variables to be continuous, as the variables are not bounded by a clear upper bound. Furthermore, our data is sequence data, meaning that it is measured over time. We are also looking for anomalies at a specific point in time, meaning

we are looking for point anomalies. However, downward-sloping or upward-sloping trends are also considered interesting events, which we will consider in the methodology as well. We do not have any data labels present, meaning that we have to use an unsupervised or semi-supervised approach, where the semi-supervised approach only works when outputs are scores instead of labels and if we can assume that the largest part of the data is normal, or if we can filter outliers from the dataset. Lastly, as described above, many multivariate outlier detection techniques consider a whole data instance (of multiple variables) an outlier. However, in our case, not all variables at a specific point in time have to be outlying. If the number of clicks on an ad is considered anomalous, the ticket sales are not necessarily anomalous. One way to resolve this is to apply univariate outlier detection methods over all these variables. However, this again loses the relation between the different variables.

To resolve this, Rousseeuw & Bossche (2018) developed DetectDeviatingCells (DDC). They first apply univariate outlier detection techniques in their method on all variables, flagging values with exceptionally high or low values compared to their column. Secondly, they use the unflagged values of the variables to predict the other variables if they are correlated. Based on these predictions and the actual values, this method can give an outlier score to each cell, which is our goal. The disadvantage is that this method does not consider the order of the data points. We know that time series in the real world can differ based on seasonality and based on previous values, therefore arriving at a missing part in the literature. Therefore, based on the literature, an extension of the DDC algorithm is required to make sure it can handle multivariate temporal data as well, which will be explained in Chapter 4.

Chapter 3

Data

In the current chapter, I will first discuss how the data has been acquired, after which I will give descriptive statistics and some insights into the data. The dataset comes from KLM, a major Dutch airline.

3.1 Data acquisition

The data used in the research comes from three different sources. First, the data comes from Search Ad 360 (SA360), a search management platform from Google, in which marketing campaigns can be managed. Data from this source include the number of impressions on an ad, the number of clicks, the cost per click, and the device. Additionally, this data source gives information about the ad group, the campaign, and the account. Secondly, the data comes from the Google Analytics (GA) platform, where more insights are given into the conversions, bounces, and qualified visits to the Electronic Booking Tool (EBT). Lastly, data comes from the Electronic Booking Tool Management Insights (EBTMI). This is the summary of the payment data from KLM per transaction. Therefore, oddly, QualifiedEBTVisits does not come from the EBTMI database, though EBT is in the name. How the data is combined and matched is shown in Figure 3.1.

To start with SA360, there are different aggregation levels of search terms and advertisements, namely keywords, ad groups, campaigns, and accounts. These keywords are not generated by hand but are also based on the search behavior of users and algorithms to maximize visibility. In total, the dataset contains 132.940.938.632 keywords in a database of over 50Tb, some of which may be duplicates, in Google Cloud Platform (GCP). This matrix is very sparse, which means that many keywords do not have any impressions or clicks, nor do they have any sales

Variable	**Source**	**Explanation**
QualifiedEBTVisits	Google Analytics	Number of visits to the website where the flight has been selected, before auxiliaries
Sessions	Google Analytics	number of times users return to the website (user can have more than one session)
Bounces	Google Analytics	number of visits with only one page interaction (unqualified visits)
Users	Google Analytics	Number of unique users on your website
QualifiedVisits	Google Analytics	Number of visits that view more than one page before leaving
QualifiedNewvisits	Google Analytics	Number of new qualified visits
QualifiedReturningVisits	Google Analytics	Number of returning qualified visits
Clicks	Search Ad 360	Number of clicks on paid search advertisements
Spend	Search Ad 360	Amount spent on paid search advertisements
Impressions	Search Ad 360	Number of times an advertisement has been seen
BookedSalesIncl	EBTMI*	Amount of euros booked, including taxes
BookedSalesExcl	EBTMI*	Amount of euros booked, excluding taxes
MaterializedSalesIncl	EBTMI*	Amount of euros actually paid, including taxes
MaterializedSalesExcl	EBTMI*	Amount of euros actually paid, including taxes
Bookings	EBTMI*	Number of tickets booked
Return on ad Spend (ROAS)	Calculated	MaterializedSalesIncl / Spend
Conversion Rate	Calculated	Bookings / Clicks
Average Order Value	Calculated	MaterializedSalesIncl / Bookings
Spend per Click	Calculated	Spend / Clicks

Table 3.1: Overview of numeric variables, including explanation.

** = Electronic Booking Tool Management Insights*

in many observations. This is because this table is updated daily, and each day all possible keywords with all their statistics are appended to the table. In total, there are 3.247.061 unique active keywords in the dataset, but as there are many other aggregation levels, the dataset contains 557.335.662 rows (this is due to daily updates, but also because the keyword on a desktop is separate from the keyword on a mobile phone). Because of this, the choice of aggregation level is important, not to have too few observations for anomaly detection, but also not too many, losing information. For example, I will consider the number of clicks on a specific day in the Netherlands, instead of each keyword separately. I will elaborate on the aggregation level later.

I will combine the data described above with the website data. When a user eventually clicks on an ad, several other interesting KPIs measure whether an ad is successful. Namely, when a user clicks on an ad but then directly goes back to the search engine (a bounce), this click is not very valuable but is paid for. However, the click will be precious when an actual booking is made. The website data knows through which channel a customer reached the website (so I know if it is through a paid search advertisement) and also which keywords were used in the search, combined with an ad group ID and a campaign ID. This way, it can be combined with the previously mentioned data source. This dataset does not contain clicks or impressions but whether a visitor is new, whether the visitor goes to a booking page, whether it is a bounce, and from which source the visitor came (different search engines such as Google or Bing for example).

Third, I also want to know if these visitors eventually buy tickets. For this, the third dataset has to be exploited, which contains all the ticket sales allocated to the paid search channel. These ticket sales are matched to the previously mentioned dataset by a visitor ID, adding additional information to the second dataset. If someone booked a ticket, they do not necessarily have to pay this directly, as they can go to a KLM office or the airport to pay for a ticket eventually. Once a ticket is paid it is called Materialized sales. Both of these values are presented with and without taxes.

Fourthly, there is data from several different search engines, which is in a different table than the keyword table but does contain similar information. This is also combined with the aforementioned datasets.

Lastly, I create extra variables that are relevant to the paid search team of KLM. I create Return On Ad Spend (ROAS) by dividing the materialized sales by the Spend variable. I furthermore create conversion rate, which is the number of bookings over the number of clicks, the average order value, which is the materialized sales over the number of bookings, and spend

per click, which is spend over the number of clicks. Note that theoretically, no division by zero error is possible, as the data contains only the paid search data, and there can be no sales without a click, and each click is paid for, corresponding to the Spend variable. It is possible to have zero over zero, which I set to zero. In some cases, because of a mistake in the attribution of the Spend variable, a division by zero error can occur. After consideration with the experts of KLM, these variables where a division by zero error occurs are also set to zero, as this mostly happens in smaller points of sale and are, therefore, not very interesting anomalies.

The tables are joined based on campaign ids, visitor ids, ad group ids, dates, search engine names, etc. All tables mentioned above have overlapping id fields on which the join can be performed. In Table 3.2, an overview of all variables is given.

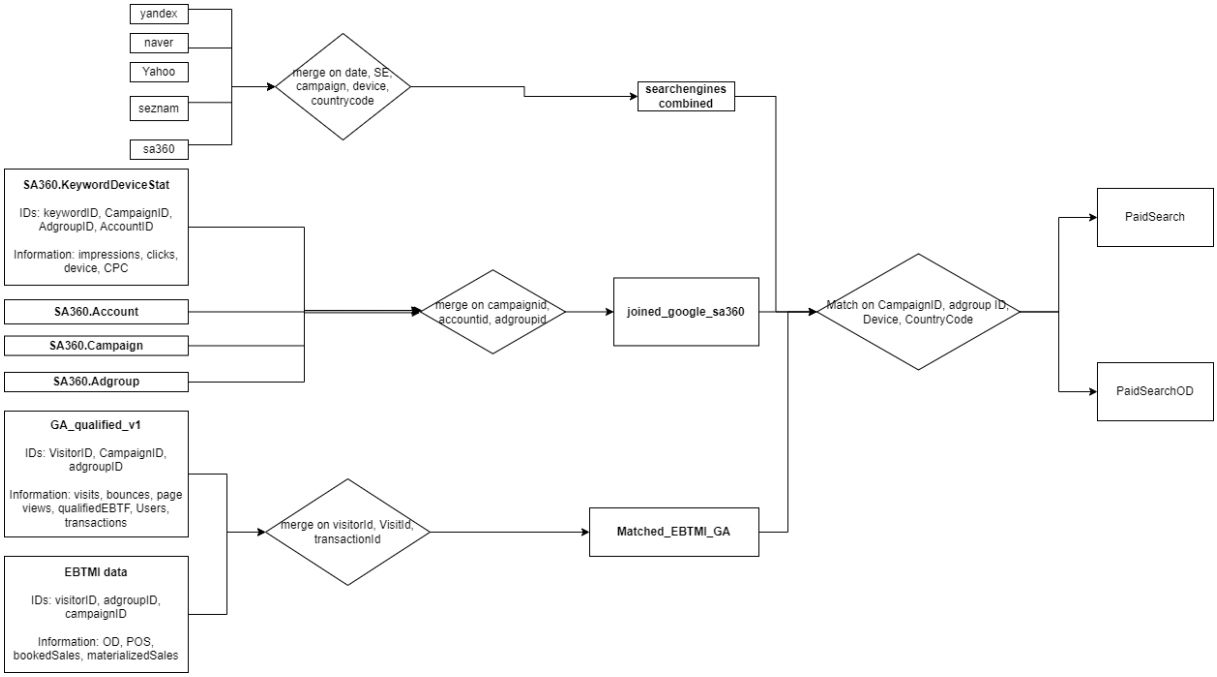


Figure 3.1: Data collection flowchart

In the research, I will experiment with different aggregation levels. The data with the highest granularity contains information at an extremely deep level, namely the variables as shown in Table 3.2 per keyword, per date, per tactic, per device, and some more aggregation levels. As explained above, this results in millions of rows with extremely many zeros. Therefore, I want to combine these data at a certain aggregation level before analyzing them. I want to investigate both on the point of sales (POS) level, which is a market level. Generally, POS is the country where the sale is made, however, in some cases, they combine certain countries into one market to have enough data to optimize their paid search strategy. The origin-destination level can

be more challenging to investigate, both because of the sparsity of the matrix, as well as to connect certain keywords to origin-destination tickets. If someone searches for 'cheap tickets' or an airline brand, and through this method, someone goes on the site and books a ticket from Amsterdam to Milan, the keyword 'tickets from Amsterdam to Milan' or its corresponding ad group does not get an impression, nor a click, but does get a booking. This attribution problem is something that KLM is working on, resulting in an origin-destination dataset that is not yet correct, and therefore not yet interesting to investigate. However, I want to create a model that can handle this format as well, such that experts within KLM can turn the model on when the data is ready. As mentioned, paid search optimizes its strategy based on POS, but also on Tactics level. KLM has nine different Tactics, which can ultimately be divided into Branded search and Non-branded search. Branded search means a search term where KLM is in the query and non-branded, for example, is a specific flight. In the Non-branded search, most of the variation is present, and the outlier detection method is most relevant. They can tweak their bidding strategy based on these levels. However, other aggregations are possible, and I want to create my model in such a way that my model can handle these kinds of aggregations.

In the collection of the data, some filters are applied. As my goal is to classify anomalies in the future in normal years, the data from 2020 and 2021 will not be used, as still many restrictions are in place because of COVID-19. This hugely affected the airline industry, resulting in non-typical observations in these years, on which I do not want to base my models. As there is no data available before 2018, there is a large difference between 2019 and 2022, and as COVID was already present in 2019 as well, I decided to only use the data from 2022, which restricts the possibility of seasonal effects, which I will discuss in the limitations.

After speaking with experts from KLM, the main focus lies on outlier detection on Point of Sale and Tactics level, as this data is most correct and has enough data points to investigate. KLM is active in 93 points of sale, but they only advertise in 65 points of sale. I therefore only include these 65 in the analysis. Furthermore, as daily data fluctuates too much, the main focus will be weekly and monthly data.

If there are missing (or NA) cells for the aggregation level, this means that there was no observation for this aggregation level, implying that it was zero. For example, if Non-branded of Nigeria in the week of 07-03-2022 has 'NA' materialized sales, there were no materialized sales in that whole week. I will therefore fill in missing values with zero, as I know why it was missing.

3.2 Descriptive statistics

I will give some insights into the data concerning Point of Sale and Tactic aggregation level. As described, I have 65 active points of sale in which paid search advertises, mainly based on country codes. Furthermore, there are two tactics, one being branded and one non-branded, yielding 130 observations across the aggregation level. I have 365 observations for each aggregation level, and 21 variables are included in the data, yielding 996.450 data entries. Table 3.2 gives an overview of the distribution. Note that all variables are greater than or equal to zero and are skewed to the right. Furthermore, notice that some outlying values can already be spotted, as the conversion rate should be lower than or equal to one, as every sale has to be connected to a click. Furthermore, a ROAS of almost 800.000, though desirable if actually true, is unrealistic. This would imply that spending one euro returns 800.000 euros. This already shows that anomalies are present in the data.

	Mean	Std	Min	25%	50%	75%	Max
AverageOrderValue	622.45	713.36	0.00	0.00	541.77	924.10	30981.80
BookedSalesExcl	17414.47	54836.04	0.00	0.00	3018.54	12137.04	852163.25
BookedSalesIncl	20230.26	63310.77	0.00	0.00	3512.33	14032.31	958803.89
BookedTickets	32.96	111.04	0.00	1.00	5.00	20.00	1614.00
Bookings	21.33	70.10	0.00	0.00	3.00	13.00	1071.00
Bounces	99.07	281.38	0.00	6.00	25.00	85.00	4340.00
Clicks	1041.52	2403.13	0.00	84.00	330.00	1036.00	33493.00
ConversionRate	0.06	0.52	0.00	0.00	0.01	0.03	42.00
Impressions	14070.94	39262.11	0.00	509.00	1802.00	8450.50	545408.00
MaterializedBooking	19.95	68.54	0.00	0.00	2.00	11.00	1061.00
MaterializedSalesExcl	15614.60	53011.98	0.00	0.00	1805.00	9286.79	826870.39
MaterializedSalesIncl	18181.81	61254.20	0.00	0.00	2113.38	10767.49	931394.30
QualifiedEBTVisits	245.58	706.98	0.00	13.00	54.00	189.00	10666.00
QualifiedNewVisits	425.90	862.23	0.00	41.00	132.00	432.00	8902.00
QualifiedReturningVisits	537.56	1695.92	0.00	31.00	118.00	363.00	25852.00
QualifiedVisits	963.46	2490.30	0.00	79.00	262.00	824.75	34341.00
ROAS	442.93	7256.48	0.00	0.00	18.60	202.12	799839.30
Sessions	1279.90	3341.58	0.00	105.00	340.00	1076.00	48991.00
Spend	261.71	846.96	0.00	13.58	44.35	170.88	21235.71
SpendPerClick	0.25	0.23	0.00	0.08	0.15	0.37	3.48
Users	860.04	2085.50	0.00	79.00	254.00	797.00	26480.00

Table 3.2: Descriptive statistics of the used variables. Note that they all have 130 aggregation levels and 365 days, yielding 47.450 data points

Chapter 4

Methodology

As described in the data section, I have three different dimensions in the data. First, there is the observational dimension. These include impressions, clicks, bounces, booked sales, materialized sales, etc. Secondly, there is the aggregation dimension. This dimension can be the point of sale (the location where the search is done), but it can also be origin-destination (like Amsterdam-New York or NL-US) or something else. Lastly, there is a time dimension. All previously mentioned variables have a timestamp when they are observed. Each variable in the observational dimension can individually be outlying in both the time dimension and the aggregation dimension. This means that a variable can be an anomaly based on time and aggregation at a certain point in time. Therefore, I can observe x_{ijt}, which is the value of variable j in country i at time t. An example is that the number of clicks in Germany Non-branded in week 40 of 2022 can be an anomaly, while the number of impressions is not considered anomalous.

Rousseeuw & Bossche (2018) created a method capable of detecting cellwise outliers, meaning that not a whole row has to be an outlier but only one variable. In their paper, they use the Top Gear dataset (Alfons, 2021) as an example, which contains numerical variables on 297 different types of cars. They show that considering bivariate relationships in anomaly detection increases the number of anomalies found and gives more insight. Though their method proved successful, their method is two-dimensional, like multivariate observations over time, and their method does not consider the time series' structure. In our approach, I will extend the work of Rousseeuw & Bossche (2018) and offer a solution to the two aforementioned issues with the method. I apply the method of Rousseeuw & Bossche (2018) on a specific date, after which I extend their detected anomalies such that they also include univariate time series anomalies, based on the approach of Maronna (2017). The reason I use a univariate outlier detection technique is that I want cell-wise outliers, and multivariate time series outlier detection methods generally only

detect whole data instances as outliers instead of only one cell.

The methodology is divided into several parts. First, I explain the data preparation, after which the DDC algorithm is reviewed and explained, then the method to extend their work is presented.

4.1 Data preparation

For applying the DDC algorithm, note that I select the points with the same timestamp depending on the aggregation level. As the authors of DDC note, the algorithm should be approximately Gaussian in its center. A popular way to transform positive data to near normality is by using the Box-Cox transformation. However, as noted by Raymaekers & Rousseeuw (2021), Box-Cox transformations are usually done by applying maximum likelihood and are, therefore, very sensitive to outliers. They propose a robust way to apply the Box-Cox transformation by reducing the effect of outlying observations. This is done by optimizing a robust criterion instead of a regular maximum likelihood estimation based on the Tukey bisquare ρ-function. Using the estimate from the robust criterion, they use a reweighted maximum likelihood estimate to eventually transform the data, where the weights of outliers are zero. The weights are determined based on a hard rejection rule based on the z-value corresponding to 99.5%. As I have some zero entries but otherwise strictly positive entries, I robustly transform the data as follows:

$$\tilde{X}_{jt_0} = \text{robBC}(X_{jt_0} + 1), \quad \forall j \in J, \tag{4.1}$$

with $\tilde{X}_{jt_0}$ a vector with the robustly normalized values of X_{jt_0}, t_0 being the date that is analyzed, and robBC being the robust Box-Cox transformation as introduced by Raymaekers & Rousseeuw (2021).

For the time series, as there are many different aggregation levels and variables, potentially creating many different time series, not all time series can be analyzed and created by inspecting autocorrelations and partial autocorrelations, but one method to analyze all of them should be considered. I will cover the preparation of the time series later.

4.2 Detect Deviating Cells

Note that I have x_{ijt}, as described earlier. I use the DDC algorithm to detect outliers across i and j. I, therefore, fix the time t, implying that I am looking for outliers at one specific time. For ease of notation, I will therefore write x_{ijt} as x_{ij}. I then get a matrix on which the DDC

algorithm can be applied. The DDC algorithm consists of eight steps, which I will discuss at a higher level. For more details, I refer to Rousseeuw & Bossche (2018).

First, the algorithm creates robust location and scale estimates and standardizes them accordingly. I calculate

$$m_j = \text{robLoc}_i(x_{ij}) \quad \text{and} \quad s_j = \text{robScale}_i(x_{ij} - m_j), \tag{4.2}$$

with m_j the robust location estimate for column j, and s_j for the scale. These estimates are similar to the first step of an algorithm for the M-estimator. Afterward, the data is standardized by

$$z_{ij} = (x_{ij} - m_j)/s_j. \tag{4.3}$$

The standardized data is used for univariate outlier detection, where z_{ij} is considered an outlier if the absolute value lies above a certain threshold c. I take $c = \sqrt{\chi^2_{1,.995}}$, because, as I have explained, I determine outliers from two different angles: the time dimension and the dimension explained above. I therefore halve the rejection region while having two models that can detect outliers. I then get

$$u_{ij} = \begin{cases} z_{ij} & \text{if } |z_{ij}| \leq c, \\ \text{NA} & \text{if } |z_{ij}| > c. \end{cases} \tag{4.4}$$

Afterward, in the next step, robust correlations are calculated between each pair of variables (z_{ij}, z_{ih}). The robust correlations are based on the robust scales and a tolerance ellipse around the two variables originally introduced by Gnanadesikan & Kettenring (1972), defined as

$$\hat{\rho}_{jh} = \left((\text{robScale}_i(z_{ij} + z_{ih}))^2 - (\text{robScale}_i(z_{ij} - z_{ih}))^2 \right) / 4. \tag{4.5}$$

This tolerance ellipse determines which points are used in calculating the correlation. A regular Pearson correlation is calculated between the points in the ellipse.

If the absolute value of the robust correlation cor_{jh} between the two variables is higher than some correlation limit (which in my case is 0.5), one can conclude that the two variables are connected, implying that the variables contain useful information about each other. For the pairs that satisfy this condition (which are called neighbors), a robust slope b_{jh} without an intercept is calculated. This robust slope is a plain least square regression line with some values filtered out based on an initial robust prediction. The values with residuals above a certain threshold are not used in the ordinary least square regression line.

These robust slopes are then used to create a predicted value of a specific cell. Each neighbor has a prediction for a cell, namely $b_{jh}u_{ih}$, which is weighted according to its correlation with the predicted cell, including the cell itself (which has a correlation and slope equal to one).

As the high values are omitted (the $|z_{ij}|$ above a certain threshold are replaced with NA and are not used in the prediction), the scale of the entries is shrunk. As this is undesirable, deshrinkage is applied to the predictions by regressing the predicted values on the actual values, yielding a correction factor. I multiply the predictions with this robust regression coefficient to correct this shrinkage.

The sixth step is to flag the cellwise outliers, where cells are flagged as an outlier if the standardized residuals are above the same threshold c, and the seventh step is to flag row-wise outliers if the average of the standardized residuals in a chi-square function is above the threshold c. The last step is to destandardize the data, eventually getting the original x_{ij} values back.

Though the DDC algorithm works well for identifying cellwise outliers at a specific time for two dimensions, for the time dimension, one wants an algorithm that can capture the order of the series without being influenced a lot by outliers. The DDC algorithm, though robust, does not capture this order. In the next section, I will address my solution to this problem.

4.3 Outlier Detection in Time Series

I want to check whether the cell at time t is an outlier for the time dimensions. Several approaches are possible, as explained in Chapter 2. Clustering- and nearest-neighbor-based approaches have a disadvantage in that they do not consider the order of the time series and can therefore only detect global outliers. In contrast, classification-based approaches require 'normal' data instances to train on. I will therefore focus on a statistical approach using time series models. I will discuss a robust way as introduced by Maronna (2017), but before this, I will give some background. For notation, note that I here focus on univariate time series. I will therefore write x_{ijt} as y_t, as is usual for time series.

4.3.1 Background

When performing regression-based outlier detection, in an ordinary regression, one can use robust estimators like MM estimators to estimate the model, after which outliers can be detected based on the residuals of the model estimation. Assuming that the time series is stationary, an intuitive approach to robustly estimate the time series would be to assume an ARMA(p, q) model as

$$y_t = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \theta_1 u_{t-1} + ... + \theta_q u_{t-1} + u_t, \quad \forall \max(p,q) < t \leq T, \tag{4.6}$$

and to estimate the parameters $\lambda = (\Phi, \Theta, \mu) = (\phi_1, \phi_2, ..., \phi_p, \theta_1, \theta_2, ..., \theta_q, \mu)$ and the scale

σ robustly. Finding a robust estimator in a regression problem can be done using an iterative process, where regression and scale parameters are updated at each iteration. One option for robust parameter estimation is the use of the M-estimator, which makes a regression more robust by using a weighting function $\rho(x)$ to decrease the weight of vertical outliers as

$$\hat{\lambda}_{\mathrm{M}} = \arg\min_{\lambda} \sum_{t=\max(p,q)+1}^{T} \rho\left(\frac{u_t(\lambda)}{\hat{\sigma}}\right), \tag{4.7}$$

with $\rho(x)$ a loss function such as the Tukey bisquare loss function or a Huber Loss function, $u_t(\lambda)$ the residuals of the estimation of the ARMA(p, q) process, and $\hat{\sigma}$ a scale estimate. In order to obtain this estimate, a proper initial estimate for the scale parameter $\hat{\sigma}$ is necessary. One option is to use the M-estimator of scale, which is defined as the solution to

$$\frac{1}{n} \sum_{t=max(p,q)+1}^{T} \rho\left(\frac{u_t(\lambda)}{\hat{\sigma}_{\mathrm{M}}}\right) = \delta, \tag{4.8}$$

where δ is a constant, and n the number of observations. Another option is using an S-estimator, as is done in MM estimators. S-estimators are disadvantaged because they can be tuned to a high breakdown point but not to a high efficiency (Maronna, 2017). To resolve this, Yohai & Zamar (1988) introduced the τ-estimator, which has a higher efficiency than a regular S-estimator. The τ-scale is given as

$$\tau^2(\lambda) = \hat{\sigma}_{\mathrm{M}}(\lambda)^2 \frac{1}{n\delta_1} \sum_{t=1}^{T} \rho_2\left(\frac{u_t}{\hat{\sigma}_{\mathrm{M}}}\right), \tag{4.9}$$

where $\hat{\sigma}_{\mathrm{M}}(\lambda)$ is the M-scale estimator based on a bounded ρ-function $\rho(x)$, δ_1 a constant which is necessary to ensure consistency for Gaussian errors, $\rho_2(x)$ a ρ-function for the tau estimator which can differ from $\rho(x)$, and u_t the residuals of the ARMA(p, q) model with parameters λ.

If I define $\hat{y}_{t|t-1}(\lambda)$ as the optimal linear prediction of y_t based on the previous observations, I could, for example, estimate the model by minimizing the robust scale of prediction residuals $\hat{u}_t = y_t - \hat{y}_{t|t-1}(\lambda)$, and use this again in a robust approximation of the λ parameter, as described before. If I write the variance of $\hat{u}_t(\lambda)$ as $a_t^2(\lambda)\sigma_u^2$, I define

$$Q^*(\lambda) = \sum_{t=1}^{T} \log\, a_t^2(\lambda) + T\, \log\, \left(\tau^2\left(\frac{\hat{u}_1(\lambda)}{a_1(\lambda)}, ..., \frac{\hat{u}_T(\lambda)}{a_T(\lambda)}\right)\right) \tag{4.10}$$

as the log likelihood function of λ, after which I define the τ-estimator as

$$\hat{\lambda} = \arg\min_{\lambda} Q^*(\lambda). \tag{4.11}$$

For a derivation of $Q^*(\lambda)$ I refer to Maronna (2017).

However, while the above process can achieve a breakdown point of 0.5 in ordinary regressions, in AR(p) models, an outlier does not only influence the estimation at the moment an outlier takes place (say y_t) but can affect residuals until y_{t+p+1}, informally yielding a maximum breakdown point of $0.5/(p+1)$ (Maronna, 2017). As MA(q) models can be written as infinite AR models (Box et al., 1994), the BP of an ARMA(p, q) model is equal to zero. Therefore, I assume that the observed values y are a combination of an ARMA(p, q) process and an outlier process. Note that y_t itself then does not follow an ARIMA(p, q) process, as y_t is the observed value with potentially contaminated data points, which I do not want to use in estimating the model. Therefore, consider

$$y_t = x_t + \omega\xi_t^{(t_0)}, \tag{4.12}$$

where $\omega\xi_t^{(t_0)}$ is the outlier process, meaning ω corresponds to the size of the effect and $\xi_t^{(t_0)}$ the effect at time t of the outlier present at time t_0. x_t follows an ARMA(p, q) process, which, in the case of no contamination in the data, would imply that y_t follows Equation 4.6, and $x_t = y_t$ for all t. Note that this notation allows for different types of outliers. Consider level shifts (LS), which increase or decrease the mean value abruptly; additive outliers (AO), which are unexpected extreme points; or innovation outliers (IO), which are additive outliers with a smearing effect. In the case of only a level shift present at time t_0, notice that ω is the effect of the level shift, and $\xi_t^{(t_0)}$ is equal to zero when $t < t_0$ and equal to one otherwise. For only an additive outlier at time t_0, notice that ω again is equal to the effect of the additive outlier, while $\xi_t^{(t_0)}$ is equal to one only when $t = t_0$ and zero otherwise.

Note that all these outliers could be present in my dataset. Whenever a tactic is changed, countries are targeted for particular campaigns, or the organization of the data changes, a level shift could occur. Also, additive outliers can happen when something temporarily breaks, or a certain event takes place, and innovation outliers could be due to a third variable, like a major worldwide event. One can imagine that a country declaring war on another country could create an innovation outlier, with high effects the moment the war has been declared and a smearing effect afterward. After inspection of the data, I restrict the outliers to additive outliers, as these are more intuitive, and smearing effects are not that interesting to the KLM team, implying that $\xi_t^{(t_o)}$ is either one or zero. Furthermore, when assuming level shifts take place, one has to difference the data in order to detect these outliers (Bianco et al., 2001), which brings along an extra complication of doublet outliers when differencing additive outliers.

4.3.2 Robust outlier detection

As described, detecting outliers in time series is tricky. I assume that the data I observe comes from Equation 4.12. In order to estimate the model and to determine which points have a relatively high residual in the estimation, several steps have to be taken, based on the work originally introduced by Bianco et al. (2001). Note that I describe the idea of the method, and refer to the same paper for more details. To do this, I first explain how I use a τ-estimator for λ and σ in combination with robust filtering to get robust estimations of λ and σ, treating the lag order p and q as known. Afterward, I explain my process for model identification based on a robust Akaike Information Criterion. Lastly, I explain how I detect outliers once the model has been robustly estimated.

As described, above estimation of the λ parameters using τ-estimators have a BP of zero in ARMA(p, q) processes. A better method is to do such minimization with a robust filter. In this filtering step, a recursive algorithm determines whether an observation at time y_t is replaced with an observation $\hat{x}_{t|t}$, which is a combination of $\hat{x}_{t|t-1}$ and the filtered residuals, defined as

$$\tilde{u}_t(\lambda) = y_t - \hat{x}_{t|t-1}, \tag{4.13}$$

in such a way that

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + s_t \psi\left(\frac{\tilde{u_t}}{s_t}\right), \tag{4.14}$$

where u_t is the prediction residual, s_t an estimation of scale and $\psi(x)$ a scoring function, which is the derivative of $\rho(x)$, which we've seen before. Note that when residuals as defined in Equation 4.13 are small, the value of $s_t\psi\left(\frac{\tilde{u_t}}{s_t}\right)$ will be close to $\tilde{u}_t$, and $\hat{x}_{t|t}$ will be close to y_t. Note that if I replace $\hat{u}_t$ with $\tilde{u}_t$ in Equation 4.10, I get a more robust way to estimate λ through Equation 4.11, with similar logic for replacing u_t with $\tilde{u}_t$ in Equation 4.9. I, therefore, first fit a τ-estimator on the data, which results in an initial estimate, after which I apply robust filtering. Afterward, I again fit a robust τ-estimate on the filtered values of y. I call this the Filtered τ estimator or the Fτ-estimator for λ and σ.

One problem is that I also have to identify the model. Usually, determining the AIC or BIC of a model yields an indication of how many AR- and MA-lags should be used. However, the AIC is based on the likelihood function, which is sensitive to outliers. Therefore, for identifying the model, I follow a procedure also given by Maronna (2017). I first fit an AR($p*$) model, where $p*$ is created based on a robust AIC (RAIC). The RAIC is given as

$$\text{RAIC}_p = \log(\tau^2(\tilde{u}_{p+1}(\hat{\lambda}_{p,\text{rob}}), ..., \tilde{u}_T(\hat{\lambda}_{p,\text{rob}}))) + \frac{2p}{T-p}, \tag{4.15}$$

where $\tilde{u}_i(\hat{\lambda}_{p,\text{rob}})$ are filtered residuals that correspond to the Fτ-estimator, and τ the respective scale. The selected AR model is the model with the lowest RAIC, and use the robust y-values that follow from the selected model. Though Maronna (2017) recommends inspecting the ACF and PACF graphs of these values, I take a rule of thumb to select the MA and AR lags, as I cannot inspect all the time series by hand. I, therefore, calculate the regular AIC on the robust y values for each ARMA model with a maximum AR and MA lag of 5, and I select the model with the lowest AIC. For determining the ARMA order of the robust values of y I use the *auto.arima* function (Hyndman & Khandakar, 2008).

I can then compute the parameter estimators of the ARMA model, which I call $\hat{\lambda}$ and $\hat{\sigma}_u$ with now known p and q.

Based on the robust filtering and model estimation, I estimated the filtered residuals $\tilde{u}_t$. I make a set of potential outliers C if

$$|\tilde{u}_t| > 2 * \hat{\sigma}_u. \tag{4.16}$$

Note that I then have y_t and the estimated $\hat{x}_{t|t}$ as an estimate for x_t, and I can then estimate the effect size for ω for each values of $t_0 \in C$ as

$$\hat{\omega}\xi_t^{(t_0)} = y_{t_0} - \hat{x}_{t_0|t_0}. \tag{4.17}$$

I can then create a t-like test statistic as

$$U^* = \frac{\hat{\omega}\xi_t^{(t_0)}}{\hat{\sigma}_u}. \tag{4.18}$$

I clean the series of the detected outlier by replacing y_t with $y_t - \hat{\omega}\xi_t^{(t_0)}$, until no t-like statistics are present greater than the critical value. These values that we replaced are the detected outliers. I refer to multiple other usages of this method for more details (Chang et al., 1988; Bianco et al., 2001; Maronna, 2017).

As a lower temporal granularity in my dataset leads to more observations, the critical cutoff value should increase with the number of observations. If we, for example, take daily observations of the past year, I have 365 observations with much more variation than weekly or monthly observations. I, therefore, take the cutoff value based on the time aggregation. For daily observations I take a $t_{5,0.995}$, for weekly observations I take $t_{6,0.995}$ and for monthly observations I take $t_{7,0.995}$. I base these values on the fact that the standard critical values of the *robustarima* package are 3 when less than 200 observations are present, 3.5 when 200-300 observations are present, and 4 when more than 500 observations are present. I use the robustarima package in R for the implementation (Kaluzny & TIBCO Software Inc., 2021).

4.4 Preprocessing time series and parameter settings

For the preprocessing of the time series, some things should be taken into consideration. First, some choices are made based on the goal of the anomaly detection model. With sales, experts of KLM would want the model to be more sensitive to points lower than expected than points higher than expected. For example, if sales within a specific country are typically steadily around 300.000 euros per week, then one week having only 30.000 euros in sales would be a more threatening anomaly than one week of 600.000 euros. Therefore, as sales data can be skewed, I log transform the sales data to be more sensitive to negative outliers.

Furthermore, as different aggregations are possible and the granularity can become very low, I impose some restrictions on the time series before I analyze them. First of all, as time series are generally based on continuous values, I impose the restrictions that at least three unique values should be present in the time series. Furthermore, the summation of the last ten observations should exceed ten, and the last ten observations should have less than 50% zeros. Lastly, the standard deviation should be higher than 0.0001, as the t-values explode when scales are very small. I use these restrictions as my model would break if the series has no or a tiny variance and as my time series model is not made for zero-inflated series. These filters mainly filter out the time series of for example minor points of sale, where only few tickets are sold. These observations are not attractive to experts of KLM anyway, as they will not spend time on small revenues. I, therefore, do not pay a lot of attention to these restrictions, as long as the larger points of sale are included.

4.5 Trend heuristic

After feedback rounds from the KLM experts, one other anomaly is helpful to detect: the presence of a trend in the last three months. My model detects outliers at a specific time but does not include increasing or decreasing values. I include a trend heuristic by fitting a robust linear model on the robust y-values, created by the algorithm in Section 4.3.2. I determine β_{MM} for an intercept and a slope and look at the t-value. I determine a cutoff value to classify a point as an outlier, which I set to 4, as I find this works well for the data. I use the *lmrob* function of the robustbase package (Maechler et al., 2022) with the standard settings and an MM-estimator for β. The number of observations is then dependent on the time aggregation. I take the 12 most recent observations for weekly observations and the most recent 90 for daily observations. I do not attempt to calculate a trend over the monthly observations, as a regression with three

data points brings along difficulty when calculating t-values.

Chapter 5

Results

This chapter will be divided up into several sections. As the methodology explained, the algorithm can run for a specific point in time, for different aggregation levels, and for different time aggregations. The insights will be given based on the Point of Sale and Tactic aggregation, with a weekly time aggregation, as this is the main focus of KLM. I will compare several weeks distributed evenly over 2022, namely the weeks of 14-02-2022, 25-04-2022, 06-06-2022, 08-08-2022, and 12-12-2022, to give insights into the working and results of the algorithm. I will cover the three parts of the algorithm (DDC, robust time series, and trend heuristic) separately.

5.1 Data Preparation

First of all, I will shortly discuss the data preparation step. As I know that the data is skewed to the right, I expect the robust lambda parameters should be smaller than one and, after some experimentation, around zero. After performing the robust Box-Cox transformation, all lambdas follow this expectation. The results of the lambdas are presented in Appendix A for the different time aggregations. Other time periods have similar results.

5.2 Detect Deviating Cells

The purpose of the DDC algorithm is to detect univariate outliers and analyze bivariate relationships at a given time. I present insights into the DDC algorithm in two ways. First, I present the robust correlation matrix associated with the variables, second, I look at some outliers at the weekly level.

As explained, the correlation matrix is different for each time period. The robust correlation

matrix of all variables over all of 2022 is shown in Table 5.1. As we have many variables and some variables have very high correlations with each other, some are left out for the purpose of making it clearer to see. This means that the variables BookedSales, MaterializedSalesExcl, BookedTickets, QualifiedEBTVisits, QualifiedReturningVisits, and QualifiedNewvisits are omitted as they are highly correlated with other variables. It furthermore is clear that many variables are highly correlated (greater than 0.5) with each other. To see whether the correlations are stable over time the standard deviations of the correlations for the dates mentioned above are calculated for the previously mentioned dates. Generally, the correlations are stable, with a standard deviation of around 0.05. Only average order value differs largely over time. As the predictions are a weighted average over all variables, I assume that this does not influence the predictions too much. The standard deviations of the correlations are presented in Appendix B.

Table 5.1: Robust correlations between the variables for weekly data in week 50 of December 2022. Àbsolute correlations above 0.5 are marked in bold

	AverageOrderValue	Bookings	Bounces	Clicks	ConversionRate	Impressions	MaterializedSalesIncl	QualifiedVisits	ROAS	Sessions	Spend	SpendPerClick	Users
AverageOrderValue	1.00												
Bookings	0.43	**1**											
Bounces	0.36	**0.65**	**1**										
Clicks	0.38	**0.8**	**0.8**	**1**									
ConversionRate	0.14	**0.64**	0.09	0.21	**1**								
Impressions	0.25	0.39	**0.67**	**0.76**	-0.18	**1**							
MaterializedSalesIncl	**0.68**	**0.95**	**0.64**	**0.81**	**0.57**	0.4	**1**						
QualifiedVisits	0.45	**0.88**	**0.81**	**0.97**	0.32	**0.67**	**0.88**	**1**					
ROAS	**0.58**	**0.67**	0.18	0.29	**0.78**	-0.2	**0.76**	0.41	**1**				
Sessions	0.44	**0.88**	**0.8**	**0.98**	0.3	**0.69**	**0.87**	**1**	0.39	**1**			
Spend	0.31	**0.59**	**0.77**	**0.89**	-0.01	**0.92**	**0.58**	**0.83**	-0.05	**0.84**	**1**		
SpendPerClick	-0.18	-0.36	0	-0.16	-0.43	0.39	-0.33	-0.18	**-0.65**	-0.16	0.29	**1**	
Users	0.43	**0.85**	**0.83**	**0.98**	0.26	**0.72**	**0.85**	**0.99**	0.34	**1**	**0.87**	-0.13	**1**

Secondly, we present weekly outliers.

The results of the DDC algorithm are shown in a Cell Map (Rousseeuw & Bossche, 2018) in Figure 5.1, where red cells mean higher than predicted and blue cells mean lower than predicted. The y-axis contains the aggregation levels and the x-axis contains the variables. A few things stand out which I will discuss. First, weeks 32 and 50 have many more outliers than the other weeks. After analysis, it was the case that in week 32 there was a problem with branded search, which was turned off for almost all countries, leading to all variables of the branded search coming from the GA database being much lower than they should have been. To understand how this resulted in the output in the table, one should notice that branded search is more than 80% of the sales of KLM. Normally, this ratio is also maintained in the clicks, impressions, and the other GA variables, implying that 80% of the clicks normally come from the branded search. When the branded search broke down, the GA variables became very low, while the sales variables remained high (as they came from a different source). This led to the DDC algorithm determining that it was normal to have low GA variables with high sales variables and therefore determined sales in non-branded search as low in comparison to branded search.

For week 50, after an analysis of the data, I found that there is a mistake in mapping clicks, impressions, and bounces of some Branded Points of Sale, implying that these variables were much lower than they should be because a new data structure was being used and was not yet correctly incorporated. This caused the number of clicks to fall, which resulted in a *ConversionRate* of more than 0.5. This was a problem that the experts at KLM did not yet know, and are working on a fix.

Thirdly, notice that variables in the Netherlands are often flagged as higher than expected. This is the case because the Netherlands is the home country and main market of KLM, yielding higher sales than expected of a country this size. One last insight is the materialized sales for Nigeria, which is lower than expected in 4/5 weeks. After investigating, this is the case because in Nigeria still, many bookings are made through offices, meaning that people do book the tickets but only pay when they arrive at the airport. This leads to lower materialized sales than booked sales, which the experts of KLM were aware of, but which they deemed to be a correct outlier.

Generally, the output of the DDC algorithm seems to make sense and was useful for the experts of KLM.

Date	# Outliers	max t-val	min t-val	# invalid
Week 17	100	31.40	-175.85	133
Week 23	137	52.20	-22.53	169
Week 32	408	22.05	-17.87	151
Week 50	94	4316.39	-58.84	166

Table 5.2: Overview of outliers on a weekly basis for different weeks in 2022

5.3 Time Series Analysis

For the time series, as the estimations are done on individual univariate time series and not on bivariate relationships, I will give insights in the working of the time series method according to an individual example. I select the weekly observations of clicks in Germany based on the branded tactics. The time series is shown in Figure 5.2. For the robust approach, the first fit based on the RAIC in Equation 4.15. This criterion selects five AR-lags, and detects three different outliers. The cleaned time series is shown as red dots in Figure 5.2. Note that week 32, and weeks 1 and 2 in January 2023 are considered outliers. Based on the cleaned values, we select an ARMA(1,1) model, after we again fit the robust model. The robust model with the new identification parameters yields the same outliers as the AR(5) model, and thus week 50 is not an outlier.

Next, I run the model for the different dates. The results of the time series analysis are shown in Table 5.2. Note that mostly the number of detected outliers is similar for the different dates, except for week 32. Here, as explained, branded search broke down, and many time outliers were found with negative t-values. 82.6% of the outliers in week 32 were outliers with a negative t-value and tactic branded. Furthermore, notice that the maximum t-value in week 50 is very high, which is because the conversion rate increased from a steady time series around 0.02 to 40, leading to a very high t-value, for reasons described in Section 5.2.

5.4 Trend Heuristic

A similar table as in previous section is shown for the trend heuristic in Table 5.3. Notice that more trend outliers were found in week 50, as some variables suddenly decreased as described earlier.

Date	# Outliers	max t-val	min t-val	invalid
Week 17	338	57.76	-20.07	133
Week 23	484	32.01	-18.41	169
Week 32	492	11.27	-35.40	151
Week 50	693	20.77	-44.08	166

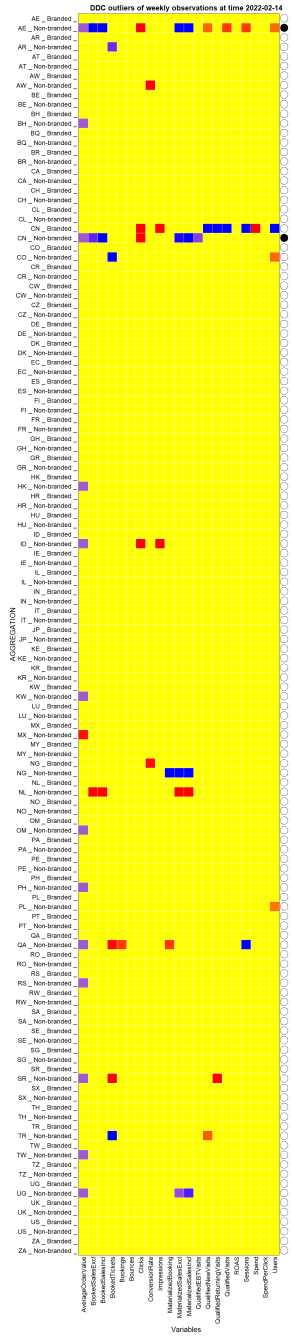
Table 5.3: Overview of weekly trend outliers for different weeks in 2022

	# Observations	# obs per model	# Outliers	# invalid
Time Series daily	1.023.750	375	31	501
Time Series Weekly	139.230	51	94	166
Time Series Monthly	35.490	13	362	6
Trend daily	1.023.750	2730	926	501
Trend weekly	139.230	2730	693	166
DDC daily	2730	-	85	0
DDC weekly	2730	-	117	0
DDC monthly	2730	-	113	0
Full Model Day	1.023.750	2730	1009	498
Full Model Week	139.230	2730	611	188
Full Model Month	35.490	2730	437	6

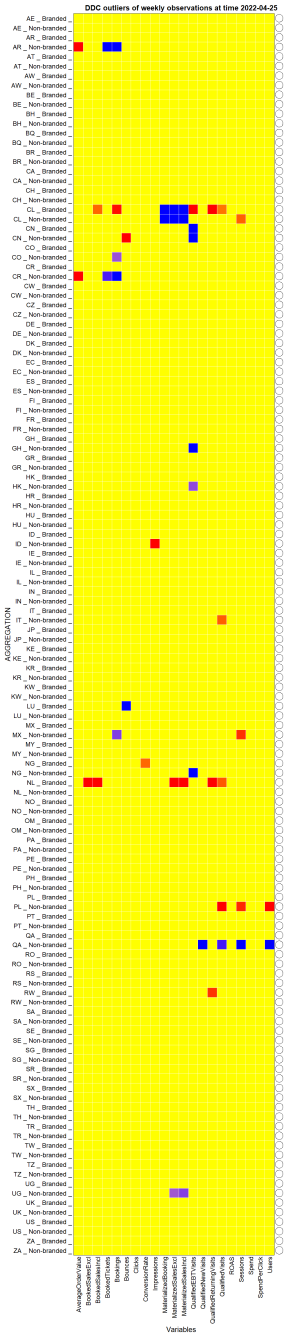
Table 5.4: Overview of the components of the anomaly detection method from data of week 50 in 2022 for different time aggregations.

5.5 Complete model

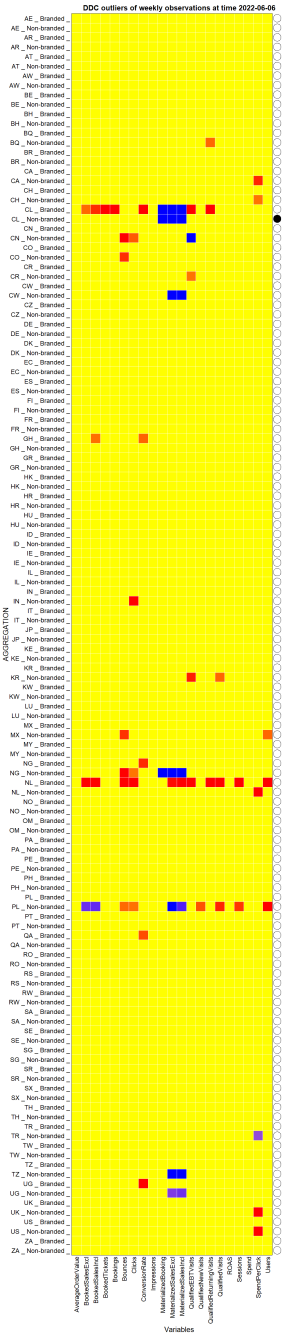
For the complete model, I take week 50 and show how the model detects outliers at different time aggregation levels. The results of this are given in Table 5.4. Note that the number of classified outliers is not the sum of the individual components of the algorithm, as the components can classify the same point as outlying. Furthermore, For the time series, as expected, the higher the granularity, the less valid time series can be analyzed, as more zeros are present in the data. Furthermore, the number of outliers in the observations with high granularity (daily data) is lower than the number of outliers with lower granularity. This could be because of the aforementioned reason, but also because of the critical t-value that increases.



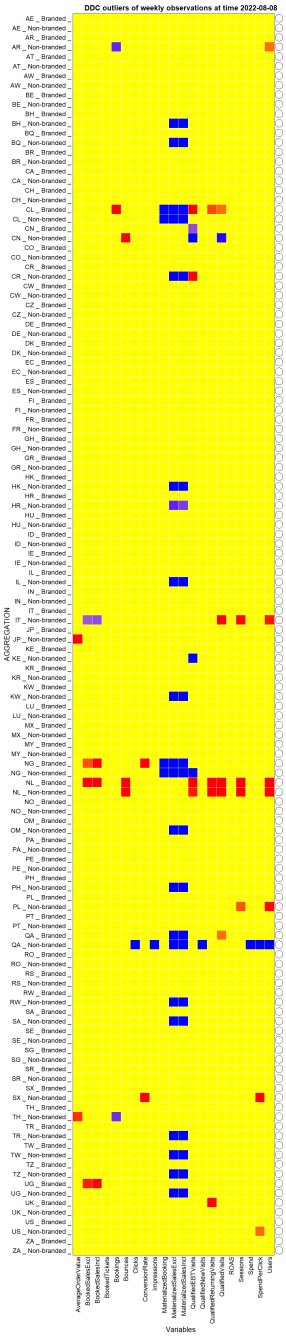
(a) Week 7



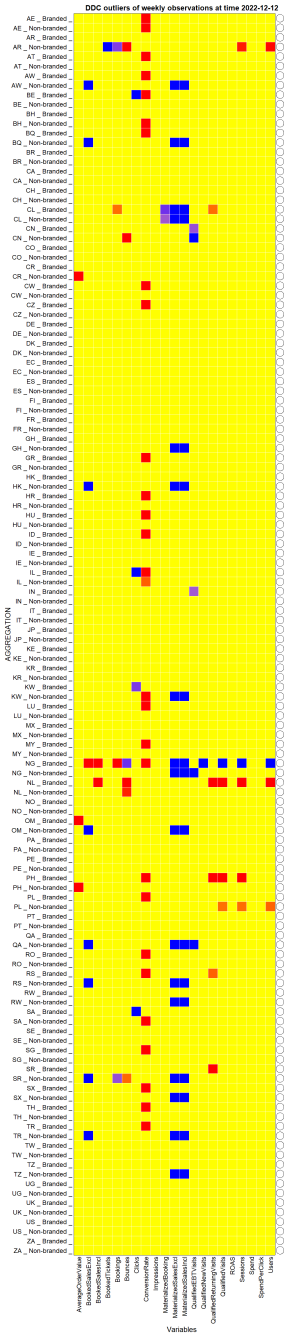
(b) Week 17



(c) Week 23



(d) Week 32



(e) Week 50

Figure 5.1: DDC results of weekly observations for different time periods

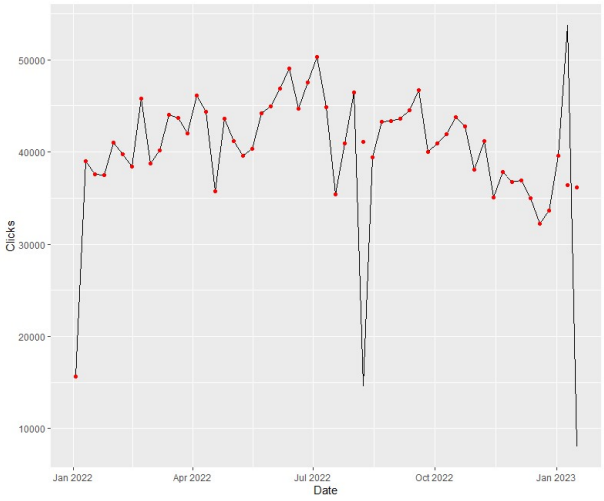


Figure 5.2: Weekly observations of the number of Clicks in Germany in Branded search, with in red the cleaned values

Chapter 6

Simulation study

When creating semi- or unsupervised anomaly detection algorithms, a difficult challenge is to evaluate said models. As there is no information on which data point is an anomaly and which is not, traditional evaluation metrics such as ROC, F1-scores, or accuracy cannot be created. My model is no different, as I have no information about which points are outliers and which are not. Based on my assumption in Section 4.3.2 that my data comes from a data generating process as in Equation 4.12, we can perform an evaluation based on simulating these kinds of time series. However, as we also want to evaluate the DDC algorithm, we need the correlation between the different time series. I perform a simulation study to evaluate the model and to determine its effectiveness. I first explain the way I simulate the data, after which the results with different parameters are given in

6.1 Methodology

For the simulation study, I use the weekly aggregation level for tactic and point of sale. For this aggregation level, I identify and calculate the AR and MA parameters described in Section 4.3.2 for each J variable. Afterward, I take the robust correlation matrix created based on Equation 4.5. I use the estimated AR parameters to simulate a time series, and I simulate errors based on a multivariate normal distribution MVN(μ_i, Σ_i) where μ_{ij} is zero for every $i \in I$ and $j \in J$ and

$$\Sigma_{ijh} = \begin{cases} \hat{\sigma}^2_{iju} & \text{if } j = h, \\ \hat{\rho}_{jh}\hat{\sigma}_{ihu}\hat{\sigma}_{iju} & \text{if } j \neq h, \end{cases}, \quad \forall i \in I \ , \tag{6.1}$$

where $\hat{\sigma}^2_{iju}$ is the Fτ-estimator of scale of the filtered residuals $\hat{u}_{ij}$ and $\hat{\rho}_{jh}$ the robust correlation between variable j and h. This generates error terms with a similar estimated scale

as the assumed DGP, which are correlated amongst variables. I afterward change the level of the time series to the median of the original variable (med(xij), which is not so important for the time series evaluation but is important for evaluating the DDC algorithm as we compare variables across countries. I then can simulate a new 'normal' dataset, which I can contaminate with outliers.

I will contaminate the multivariate time series at time t with additive outliers by randomly modifying data points with a random normal distribution as follows:

$$\xi_{ijt} = \begin{cases} y_{ijt} & \text{if } o_{ijt} = 0, \\ \mathrm{N}(y_{ijt} + (2 * v * \mathrm{Bernoulli}(q) - v) * \hat{\sigma}_{iju}, \hat{\sigma}_{iju}) & \text{if } o_{ijt} = 1. \end{cases} \tag{6.2}$$

Here, o_{ijt} follows a random Bernoulli process that determines the proportion of outliers, and v, which I call the contamination constant, is a constant that I can vary. I will set the number of outliers at 10%, 20%, and 30% to evaluate the effect of the contamination levels. Note that the expression $2 * v * \mathrm{Bernoulli}(q) - v$ generates $-v$ or v with probability q, implying that the outliers will follow a random distribution with the same scale parameter as the time series around the initial y_{jt} plus or minus v times the robust scale estimate. I will set $q = 0.5$ and v equal to 3, 5, and 7.

For each parameter setting, I will generate 20 simulations per aggregation level and time series. In order to do this the covariance matrix must be positive definite in all cases. As we have highly correlated variables with different individual variances this condition does not have to hold. Therefore, I take a subset of the variables to evaluate. This means that, as I have 65 points of sale and two Tactics, I will get 130 * the number of variables * 20 time series with pollution per parameter specification.

6.2 Results

This section will first give insights into the simulation of some variables. As described, covariance matrices will be used to simulate errors from the multivariate normal distribution. The covariance matrix of branded search in Germany is given in Table 6.1. The selected variable names are written above. One simulation took approximately 15 minutes to run, implying 5 hours for the total simulation task.

I next show some examples of simulated time series and original time series to show similarities and differences with the original data. I will again use the number of clicks in Germany in the example. The simulated time series and the original time series are shown in Figure

	AverageOrderValue	Clicks	MaterializedSalesIncl	QualifiedVisits	ROAS	Spend	SpendPerClick
1	66.10	124.64	1.11	216.36	29.14	37.80	-0.09
2	124.64	3405.04	9.67	2407.76	-10.07	1068.35	0.31
3	1.11	9.67	0.10	14.76	0.95	2.46	-0.01
4	216.36	2407.76	14.76	3238.93	169.98	703.54	-0.86
5	29.14	-10.07	0.95	169.98	51.45	-27.53	-0.24
6	37.80	1068.35	2.46	703.54	-27.53	375.95	0.57
7	-0.09	0.31	-0.01	-0.86	-0.24	0.57	0.01

Table 6.1: The covariance matrix of a selection of variables for Germany in week 50

	# Simulations	# Outliers	Precision	Recall	F1-score
v=3, out=10%	18.200	1681	0.81	0.32	0.46
v=5, out=10%	18.200	1681	0.85	0.72	0.78
v=7, out=10%	18.200	1681	0.86	0.87	0.87
v=3, out=20%	18.200	3311	0.87	0.24	0.38
v=3, out=30%	18.200	4938	0.91	0.14	0.24

Table 6.2: Simulation results while varying contamination constant and contamination percentage

6.1. The parameter settings are $v = 3$, and 10% outliers. Something interesting to notice is that the visual outliers in the original time series seem to deviate even further from the other observations than the outliers in the simulated time series, implying that the outliers in the simulation can be 'underestimated' in comparison to the actual outliers. Visual inspection of the time series without knowing the labels would therefore also not clearly label these points as outlying. A value of $v = 5$ might make outliers clearer and should make the performance of the model better.

The simulation results are shown in Table 6.2. As expected from the example, more observations created by the outlier DGP are classified as inliers than the other way around. Furthermore, the precision of the model remains high. This means that the points classified as outlying are generally also simulated outliers. However, a higher contamination level leads to more outliers not being detected. I leave it to future research to further investigate exactly why this is the case.

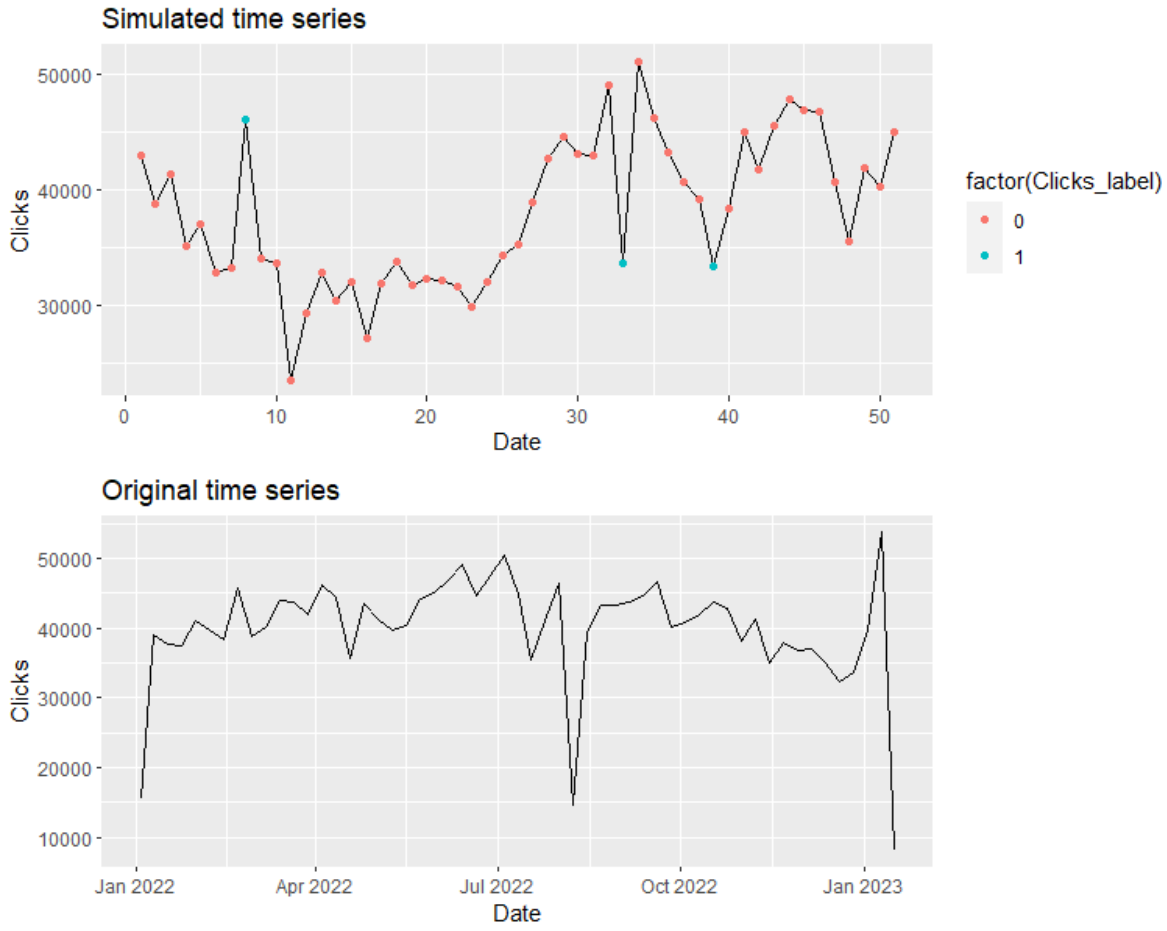


Figure 6.1: Simulated time series of Clicks for Germany with contamination of 10% with 3 standard deviations and the original time series. Outliers are marked in blue

Chapter 7

Concluding remarks

This thesis focused on outlier detection in paid search advertisements for KLM. In this chapter, I will conclude based on our results. Afterward, limitations and suggestions for future research are discussed.

7.1 Conclusion

This thesis aimed to detect anomalies in the search and click behavior of users in the airline industry. I created the dataset based on several data lakes from KLM, namely the data provided by Google Analytics, data provided by the electronic booking tool of KLM, and data based on search ad 360. Different types of outliers were defined based on the input from experts within KLM and the literature. I covered different outlier detection methods in the literature, and eventually, a combination of outlier detection procedures was chosen to use in the algorithm. The created algorithm extended the Detect Deviating Cells algorithm (Rousseeuw & Bossche, 2018) by including robust time series estimation (Maronna, 2017) and a trend heuristics, enabling it to detect cell-wise outliers both in the time dimension as well as in the aggregation dimension for each variable. The results show valuable insights that detected errors in the way KLM data was processed, resulting in a fix on the side of KLM. Furthermore, insights were given into the performance of different markets and variables. Based on the discussion with the experts, many detected anomalies were correct and could be explained by the experts implying that the model performed well. In some cases the detected anomalies were not yet on the radar of the experts, which resulted in changing bidding strategies and thereby saving costs. The simulation study shows that our model is able to accurately detect outliers for different levels of contamination and different contamination effects. Generally, larger contamination effects make it easier for

the algorithm to spot the outliers, and with higher percentages of outliers, the recall goes down, implying that the percentage of the detected outliers goes down, however, the precision remains high, meaning that the detected outliers remain actual outliers.

In this research, I focused on a specific dataset. However, the created algorithm is not limited to being used only on this dataset. Many different models and applications of anomaly detection are already present in the data, but to the best of my knowledge, this research is the first to perform cellwise outlier detection over multivariate time series. This means that my algorithm can be used to monitor different datasets in other digital marketing teams (like advertising on social media or mail) but one can imagine many other use cases for these kinds of algorithms as well.

7.2 Limitations and Future Research

The largest limitation of the research is that I only considered data from 2022. The reason for this is that the data from 2020, 2021, and parts of 2019 were contaminated because of COVID-19, and the data only goes back until 2018. This brings along complications that seasonality could not be included in the model, while experts did mention that seasonality always plays a big role. This complicates matters because of two reasons. First of all, the time series does not seem stationary in some cases, as only one year's observation is taken into account. I chose to still approach the problem as if it is stationary, as in the long term, this would probably be the best way to go (with potentially a trend). Another option would have been to differentiate the data, which has pros and cons. An additive outlier at a specific point, when differenced, creates a doublet outlier, where the next spike is symmetric around zero. One has to clean this from the series before estimating effects. However, level shifts would be easier to implement and my outlier detection method would detect level shifts in a differenced series. This could be possible to do, however, and we leave this for future work. My recommendation to KLM is to keep collecting data and reevaluate the current model in order to include seasonality and a proper trend.

Furthermore, the assumption that I made is that the data is generated by an ARMA process and is contaminated with outliers at specific points. As anyone can suspect, the sales data in the aviation industry is dependent on much more than only previous observations, namely competitors, the financial health of a country, international affairs, international events, digital marketing campaigns, etc. A real option is that only a very small part of the data-generating

process is actually because of this time series, and the largest part because of these other variables or because of white noise.

Additionally, many different aggregation levels are possible. In the created model, one can detect outliers on different time aggregations (daily, weekly, monthly), but also on point of sale, point of sale and tactic, origin-destination, etc. I restricted the results to point of sale and tactic and kept the time aggregation level mainly at weekly, as this was the main focus of KLM. It would be interesting for future research to vary these aggregation levels and see how the model performs and what kind of outputs the model gives.

Next, though our simulation method did give insights into the performance of the model, more research should be done on which kind of outliers the model missed and which outliers the model detected correctly. With 30% outliers, which was one of the scenarios, one can imagine that the algorithm has a hard time distinguishing normal and abnormal observations, but this should be backed up by actual analysis. Also, contamination percentages of less than 10% would be interesting to investigate as well as different contamination constants for different contamination percentages.

Another limitation is in the data. In this research, I considered the data as the truth, however, the allocation of sales and clicks to specific bookings is a difficult process. An example is the extreme outliers of conversion rate and ROAS, where sales are allocated to paid search while only very few clicks are measured. In some cases, these outliers consist of a large part of the observations, when the team turns off the advertising for specific Tactics or specific points of sale.

Acknowledgments

KLM

Pieter Groeneveld for enabling me to find a match within KLM, Tijmen Kort, Frie Roijers, and Maxim Volgin as Quantitative Marketing Team for providing support and insights with data-related matters, Jorin Lindenberg and Stijn Meertens as Paid Search experts within KLM who helped in determining the scope and evaluating intermediate and eventual results, Renske Siersema for helping with all HR-related problems, and Dirk de Raaff as teamlead of Quantitative Marketing.

Erasmus University Rotterdam

dr. Andreas Alfons for guiding the thesis, providing insights on different models, and providing feedback on drafts. Dr. Mikhail Zhelonkin for being co-reader and additional assessor.

Appendix A

Results of the robust Box-Cox transformation of the lambdas

Variable Name	$\hat{\lambda}_d$	$\hat{\lambda}_w$	$\hat{\lambda}_m$	$\hat{\lambda}_y$
AverageOrderValue	0.290	0.377	0.277	0.221
BookedSalesExcl	0.328	0.021	0.048	0.155
BookedSalesIncl	0.324	0.350	0.039	0.152
BookedTickets	-0.297	-0.094	0.041	0.124
Bookings	-0.351	-0.126	0.017	0.120
Bounces	-0.032	0.073	0.090	0.124
Clicks	0.135	0.219	0.274	0.129
ConversionRate	-0.162	-0.101	-0.088	-0.382
Impressions	0.176	0.210	0.244	-0.078
MaterializedBooking	-0.606	-0.230	-0.073	0.019
MaterializedSalesExcl	0.247	-0.053	0.035	0.060
MaterializedSalesIncl	0.258	-0.080	0.023	0.059
QualifiedEBTVisits	0.016	0.020	0.000	0.043
QualifiedNewVisits	0.001	0.005	-0.078	-0.027
QualifiedReturningVisits	0.019	0.040	0.038	0.019
QualifiedVisits	0.049	0.029	0.049	0.016
ROAS	-0.433	-0.213	-0.227	-0.098
Sessions	0.054	0.062	0.052	-0.000
Spend	-0.042	0.136	0.250	-0.118
SpendPerClick	-0.331	-0.325	-0.317	-0.410
Users	0.047	0.066	0.089	0.022

Table A.1: Lambda hats for the robust Box-Cox transformation

Appendix B

Element wise standard deviations of correlation matrices over time

Variable	**Average Standard Deviation**
AverageOrderValue	0.187143
BookedSalesExcl	0.049524
BookedSalesIncl	0.046190
BookedTickets	0.046667
Bookings	0.045238
Bounces	0.054286
Clicks	0.079048
ConversionRate	0.086190
Impressions	0.058571
MaterializedBooking	0.048571
MaterializedSalesExcl	0.064286
MaterializedSalesIncl	0.065714
QualifiedEBTVisits	0.038095
QualifiedNewVisits	0.039524
QualifiedReturningVisits	0.040476
QualifiedVisits	0.034762
ROAS	0.072381
Sessions	0.033333
Spend	0.066667
SpendPerClick	0.070952
Users	0.034286

Table B.1: Average standard deviations of the correlations over time for selected dates.

References

Aggarwal, C. C. (2017). *Outlier analysis*. Springer International Publishing. doi: 10.1007/978-3-319-47578-3

Alfons, A. (2021). robustHD: An R package for robust regression with high-dimensional data. *Journal of Open Source Software*, *6*(67), 3786. doi: 10.21105/joss.03786

Angiulli, F., & Fassetti, F. (2007). Detecting distance-based outliers in streams of data. , 811–820.

Bianco, A. M., Ben, M. G., Martínez, E. J., & Yohai, V. J. (2001). Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting*, *20*, 565-579. doi: 10.1002/for.768

Bianco, A. M., Martinez, E., Ben, M. G., & Yohai, V. (1996). Robust procedures for regression models with arima errors. In *Compstat* (pp. 27–38).

Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021, 6). A review on outlier/anomaly detection in time series data. *ACM Computing Surveys*, *54*. doi: 10.1145/3444690

Box, G., Jenkins, G., & Reinsel, G. (1994). *Time series analysis; forecasting and control* (3rd ed.). Englewood Cliff, New Jersey: Prentice Hall.

Braverman, S. (2015). Global review of data-driven marketing and advertising. *Journal of Direct, Data and Digital Marketing Practice*, *16*, 181–183.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). *Lof: Identifying density-based local outliers.*

Burger, M., Knaap, B., & Wall, R. (2013, 06). Revealed competition for greenfield investments between european regions. *Journal of Economic Geography*, *13*, 619-648. doi: 10.1093/jeg/lbs024

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1–58.

Chang, I., Tiao, G. C., & Chen, C. (1988). Estimation of time series parameters in the presence of outliers. *Technometrics*, *30*(2), 193–204. Retrieved 2023-02-21, from `http://www.jstor.org/stable/1270165`

Ghahremani-Nahr, J., & Nozari, H. (2021). A survey for investigating key performance indicators in digital marketing. *International journal of Innovation in Marketing Elements*, *1*(1), 1–6.

Gnanadesikan, R., & Kettenring, J. R. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, 81–124.

Goldstein, M., & Uchida, S. (2016, 4). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE*, *11*. doi: 10.1371/journal.pone.0152173

Hawkins, D. M. (1980). *Identification of outliers*. doi: 10.1007/978-94-015-3994-4

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, *22*(2), 85–126.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, *26*(3), 1–22. doi: 10.18637/jss.v027.i03

Kaluzny, S., & TIBCO Software Inc. (2021). robustarima: Robust arima modeling [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=robustarima` (R package version 0.2.6)

Laffey, D. (2007, 5). Paid search: The innovation that changed the web. *Business Horizons*, *50*, 211-218. doi: 10.1016/j.bushor.2006.09.003

Maechler, M., Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., ... Anna di Palma, M. (2022). robustbase: Basic robust statistics [Computer software manual]. Retrieved from `http://robustbase.r-forge.r-project.org/` (R package version 0.95-0)

Markou, M., & Singh, S. (2003). Novelty detection: A review - part 1: Statistical approaches. *Signal Processing*, *83*, 2481-2497. doi: 10.1016/j.sigpro.2003.07.018

Maronna, R. (2017). *Robust statistics: Theory and methods (with r)*. Wiley. Retrieved from `https://onlinelibrary-wiley-com.eur.idm.oclc.org/doi/pdf/10.1002/9781119214656`

Munir, M., Siddiqui, S. A., Dengel, A., & Ahmed, S. (2019). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, *7*, 1991-2005. doi: 10.1109/ACCESS.2018.2886457

Peck, J. (2011). Applications of outlier and anomaly detection in sponsored search advertising campaigns.

Raymaekers, J., & Rousseeuw, P. J. (2021). Transforming variables to central normality. *Machine Learning*, 1–23.

Rousseeuw, P. J., & Bossche, W. V. D. (2018, 4). Detecting deviating data cells. *Technometrics*, *60*, 135-145. doi: 10.1080/00401706.2017.1340909

Rutz, O. J., & Bucklin, R. E. (2011). From generic to branded: A model of spillover in paid search advertising. *Journal of Marketing Research*, *48*(1), 87–102.

Yohai, V. J., & Zamar, R. H. (1988). High breakdown-point estimates of regression by means of the minimization of an efficient scale. *Journal of the American Statistical Association*, *83*, 406-413. doi: 10.1080/01621459.1988.10478611

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., ... Chawla, N. V. (2019). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 1409-1416. Retrieved from `www.aaai.org`