Pontifícia Universidade Católica
do Rio de Janeiro

Rebecca Porphírio da Costa de Azevedo

A model-centric sequential approach to outlier ensembles in a marketing science context

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática, of PUC-Rio, in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Hélio Côrtes Vieira Lopes

Rio de Janeiro
September 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Rebecca Porphírio da Costa de Azevedo

A model-centric sequential approach to outlier ensembles in a marketing science context

Dissertation presented to the Programa de Pós–graduação em Informática, of PUC-Rio, in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the undersigned Examination Committee.

Prof. Hélio Côrtes Vieira Lopes
Advisor
Departamento de Informática – PUC-Rio

Profª. Simone Diniz Junqueira Barbosa
Departamento de Informática – PUC-Rio

Rafael Barbosa Nasser
Laboratório de Engenharia de Software – PUC-Rio

Prof. Gustavo Robichez de Carvalho
Departamento de Informática – PUC-Rio

Prof. Marcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, September 6th, 2018

Rebecca Porphírio da Costa de Azevedo

The author graduated in journalism from Universidade Federal Fluminense (UFF-RJ). Got certificated on R and Python programming, specialized in data mining, regression and machine learning. The author has also run studies about the ad's impact on brand affinity and sales to Facebook, and developed machine learning routines to Globo.com user's classification for websites Globoesporte.com and Cartola FC.

Bibliographic data

Azevedo, Rebecca Porphírio da Costa de

A model-centric sequential approach to outlier ensembles in a marketing science context / Rebecca Porphírio da Costa de Azevedo; advisor: Hélio Côrtes Vieira Lopes. – 2018.

78 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, 2018.

Inclui bibliografia

1. Informática – Teses. 2. Outliers. 3. Detecção de padrões. 4. Aprendizado sequencial. 5. Marketing Science. I. Lopes, Hélio Côrtes Vieira. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

To my family, without whom I wouldn't be here. To all their guidance and energy and to all the times they said I could do whatever I wanted to. To my father, who's not here anymore, but is surely proud of what I've accomplished. Thanks for shining down on me. To my husband Vilson, who's my pillar of strength and my beacon of light at all times. To all the times he's put me in the right path and to all his understanding and comprehension to all the hours I've spent away studying. Thanks for raising me up. To my sister, who's my image of compassion. To my mom, who's my role of a fighter. To professors Helio and Simone that from day one believed that even without a technical undergraduation I could go beyond and achieve something more. Thanks for always having faith and patience. To Globo.com for having started me on this journey. To Facebook and specially to Ana Cester, for all her motivation and understanding always. Thanks for being a friend. To all my friends and colleagues scattered around the globe, for all my absense and for all the times I wasn't there.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001.

Abstract

Azevedo, Rebecca Porphírio da Costa de; Lopes, Hélio Côrtes Vieira (Advisor). **A model-centric sequential approach to outlier ensembles in a marketing science context**. Rio de Janeiro, 2018. 78p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Latest years evolution in mobile devices has increased dramatically the amount of data and available information for advertisers around the world. Computational cost and available time to process data and be able to distinguish true users from anomalies or noise has only increased. Thus, the creation of a method to detect outliers could support Marketing researchers and increase their precision in understanding online behavior. Recent studies show that, so far, meta-algorithms have not been used to detect outliers. Meta-algorithms tend to bring benefits because they reduce dependency that a single algorithm can generate. This work proposes a sequential model-centric ensemble design that uses different algorithms in outlier detection to obtain better results than those obtained by a single algorithm. The novelty in this approach consists in: (i) exploring the sequential technique, using algorithms that impact the next one and whose results are a combination of previously obtained results; (ii) centralizing performance around the model and not the data, which means the ensemble is applied in the whole dataset and not on different subsamples; (iii) support Marketing researchers that need to operate data Science in a more robust and coherent way.

Keywords

Outliers; Pattern Recognition; Guided learning; Marketing Science.

Resumo

Azevedo, Rebecca Porphírio da Costa de; Lopes, Hélio Côrtes Vieira. **Ensemble sequencial centrado em modelos para detecção de outliers no contexto de Marketing Science**. Rio de Janeiro, 2018. 78p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O desenvolvimento visto nos últimos anos em dispositivos móveis tem tornado dramático o aumento na quantidade de dados e informações disponíveis para publicitários ao redor do mundo. Custo computacional e tempo disponível para processar dados e ser capaz de distinguir verdadeiros usuários de anomalias ou ruído têm crescido. Assim, a criação de um método para detecção de outliers poderia apoiar melhor os pesquisadores de Marketing e aumentar sua precisão na compreensão do comportamento digital. Estudos atuais mostram que, até o momento, o uso de meta-algoritmos tem sido pouco usado para detecção de outliers. Meta-algoritmos tendem a trazer benefícios porque reduzem a dependência que um único algoritmo pode gerar. Esta dissertação propõe um design de meta-algoritmo que utiliza diferentes algoritmos para obter resultados de detecção de outliers melhores do que aqueles obtidos por apenas um único algoritmo: centrado em modelo e sequencial. A novidade da abordagem consiste em (i) explorar a técnica sequencial, utilizando algoritmos que são aplicados sequencialmente, no qual um algoritmo impacta o próximo e o resultado final é uma combinação dos resultados obtidos; (ii) centralizar a performance no modelo e não nos dados, o que significa que o ensemble é aplicado a todo o conjunto de dados ao mesmo tempo e; (iii) apoiar pesquisadores de marketing que precisem operar ciência de dados de forma mais robusta e coerente.

Palavras-chave

Outliers; Detecção de padrões; Aprendizado sequencial; Marketing Science.

Table of contents

List of figures

List of tables

List of Abreviations

CBLOF – *Cluster-Based Local Outlier Factor*
COF – *Connectivity-based Outlier Factor*
DBSCAN – *Density-based spatial clustering of applications with noise*
ECG – *Electrocardiography*
FPR – *False Positive Rate*
IQR – *Interquartile range*
LOF – *Local Outlier Factor*
MDEF – Multi-Granularity Deviation Factor
MLE – *Maximum Likelihood Estimates*
MRI – *Magnetic resonance imaging*
ODIN – *Outlier Detection using In-degree Number*
PCA – *Principal Component Analysis*
PET Scan – *Positron emission tomography*
ROC – *Receiver Operating Characteristics Curve*
ROCK – *RObust Clustering using linKs*
TPR – *True Positive Rate*

1
Introduction

In 2017, total online retail revenue was $2.3 trillion dollars worldwide - 24.8% more than in 2016, according to the Digital Research Institute eMarketer(1)'s report. 58.9% of that, or $1.3 trillion, happened only on mobile devices (not on laptops or desktops). The same report estimates that by 2021 these numbers should nearly triple - mobile buying would represent 72.9% of any Online buying. In Brazil, total revenue was $4.73 billion dollars. And 26.4% of all online buying happened on mobile. (Figure 1.1)

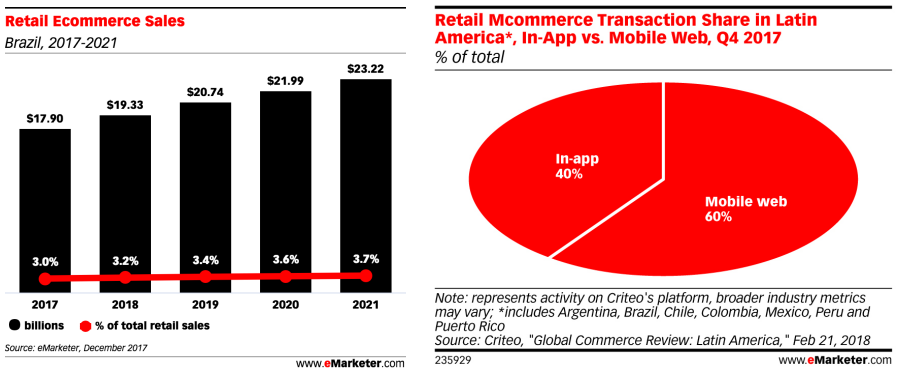

Figure 1.1: eMarketer - Worlwide Retail and eCommerce Sales.(1)

On their way to reach out to consumers and online buyers, online retailers try to influence the path to purchase by buying media - becoming an advertiser and buying ads. In Brazil, $3.89 billion dollars were spent on online ads(2) in 2017.

And one of the biggest challenges for online advertisers is to know if their investment in online media brought them any return. To be able to do this, advertisers need to start answering business questions. Bell, Corsten and Knox quote some of those (2):

(i) Who should we attribute the purchase to: traditional or online media?

(ii) How many touch points did the consumer have with the brand until finally buying?

(iii) Which publisher influenced the buyer the most?

(iv) Is my brand communicating with this buyer on different points of his decision path?

To answer all those questions you need to analyze data. And, as Smallwood(3) teaches us in his paper, this new reality of multi channel and different devices has led the task of analyzing a user's buying behavior to another level of difficulty: "the ability to capture and measure online activities hasn't come with a pocket book".

The new scenario of online buying brought advantages, but also brought challenges. Computational cost and available time to process data and to be able to distinguish real clients from anomalies or noise is growing. We need to look for new ways to analyze that make better use of available data and resources and that can be applied in the context of marketing science.

1.1 Dissertation outline

This work has been divided in four chapters, after this introductory one. Chapter 2 ("Theoretical background and revision of the state of the art") describes important concepts to understand the work proposed by this author, since "online buyers" to "ensemble learning methods". Furthermore, it exposes the techniques mentioned throughout the dissertation and describes some that are yet being studied.

Chapter 3 ("Techniques and methodologies") describes the different algorithms and components tested to build the ensemble. It also describes the method used to normalize scores and to combine ensemble models.

Chapter 4 ("Experiments and results") presents and discusses the achieved results along with technique's respective efficiencies in identifying outliers.

Finally, Chapter 5 ("Conclusion and future work") presents conclusions and discussions for future works.

One of the expected outcomes of the model is to increase observed lift in purchases when running campaigns on media and analyzing their performances. The Lift concept is nothing more than comparing results obtained when splitting an audience into Control Group and Exposed Group and comparing their performances in sales impact. The difference between Control and Exposed groups is that control group is kept from seeing the advertiser‘s creative, while the exposed groups sees the ad. Since the only difference between them is this, any results obtained from the comparison can be understood as having been caused by the ads.

Lift in Sales or Revenue is usually underestimated when the audience analyzed has a lot of outliers present. That happens because since range of product

prices can be very wide, it can also impact this comparison. Suppose, for example, Control group has had 10 purchases, but one of the purchases had a revenue 10x greater than the average revenue of all purchases. Now also suppose Exposed group had 20 purchases, but all of them regarding products of low value. When comparing revenue between the groups, one could end with a flat lift, or a very low difference between groups. And after removing outlier in revenue, for example, one could find out that ads actually had caused a significant lift.

2
Theoretical background and revision of the state of the art

This chapter presents the most important concepts that touch digital marketing, marketing science and outlier ensembling - since online shopping to ensemble categorizations. It also explains some of the main techniques for outlier detection, such as concepts that will be used to evaluate the final model.

2.1
Digital shopping and Marketing Science

2.1.1
Digital buyers and Online Retail

Figure 2.1: Nielsen states that there are three big decisions in a user's path to purchase that can be influenced by online ads. Source: (4)

Online shopper and online buyer, according to Nielsen(4), describe different phases of a user's path to purchase. The user goes from being a shopper to being a buyer when effectively buys online. (Figure 2.1) Nielsen(4) says that there are three big challenges to understand online buyers:

(i) Subtle distinctions between the buyer itself and other kinds of consumers;

(ii) Lack of focus when deciding which buying decisions can be influenced;

(iii) Lack of store-level and buyer-level tools that allow advertisers to decide where the real marketing effectiveness occurs for the buyers.

Bell et al.(5) believe that the more abstract an online buyer's objective, the more it happens what the authors define as the "unplanned buying". And during all its purchase path, the user is exposed to many touchpoints before getting to online shopping and can easily change any possible planning, the authors explain. Research institute Ipsos(6) believes that correctly understanding a user's path to purchase can even double the amount of purchases of an online retailer.

2.1.2 Cookies

Cookies are files stored in a user's computer, Oppenheimer(7) explains. These files contain encripted information and are received or updated by the user's browser almost everytime a website is accessed(7). Files are usually used to store, keep and monitor information regargind transactions performed on visited websites, just like information about the user itself(7). (Figure 2.2)

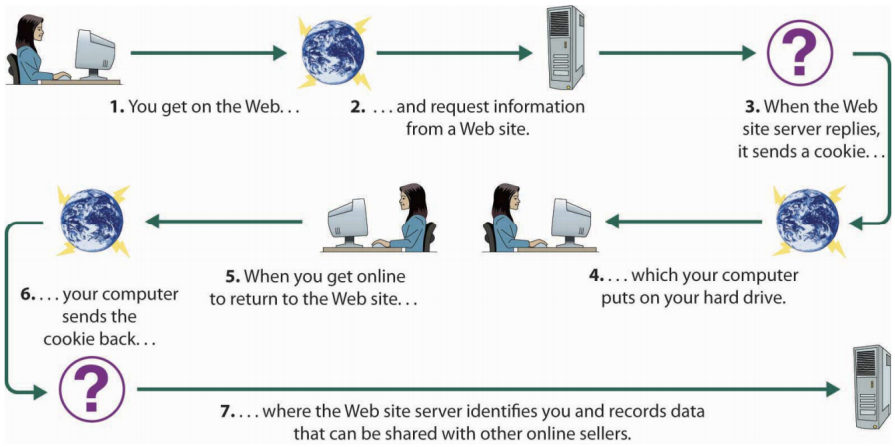

Figure 2.2: Georgia Southern University - example of how a cookie works.(8)

Cookies[1] are usually used to monitor navigation on desktops and laptops.

Howerver, smartphone data monitoring does not use cookies(7), what makes a user's path to purchase analysis even more of a challenge to the marketing industry. If a user access the same website from different browsers, it gets a different cookie from each device(7). A data analysis tool that uses cookies to monitor website's visits for a brand, tends to overestimate the number of unique

[1]According to Wikipedia in `http://en.wikipedia.org/wiki/HTTP_Cookie`, the HTTP Cookie is a file created by a web server and stored to keep user's preferences. It originates from or is sent to a different website than the one is being seen at the moment.

users it has and to underestimate the frequency these users have on this brand's website - all because it is unable to understand that two or more cookies can actually be a single unique user(7).

2.1.3 Attribution models

Zhao(9) points out that the fundamental problem when measuring online ad's efficiency is the attribution problem. Multiple digital channels, such as search, display, digital videos etc., are used for online propaganda. Each user is exposed to a combination of different channels before making the decision to buy. That's why, Zhao argues(9), to be able to correctly attribute a purchase to a group of online publishers, you have to connect the user through its entire online journey.

Most common tools for analyzing online performance attribute online purchases using the last click attribution model, according to IAB(10). This model consists in considering the last channel visited as "the cause" for a user's online purchase, ignoring all its prior navigation history. IAB(10) states that it's not possible, without the right tools, to know for sure if the purchase decision was actually made on the last click or on the first. Or even if the user had already decided on purchasing and seeing the ads had no effect on him at all. And this happens because ordinary tools can rely only on cookies to track a user's path(7) making it improbable to know if the user that saw but didn't click on an ad on a mobile device was the same that the one that actually bought the product on desktop. (Figure 2.3)

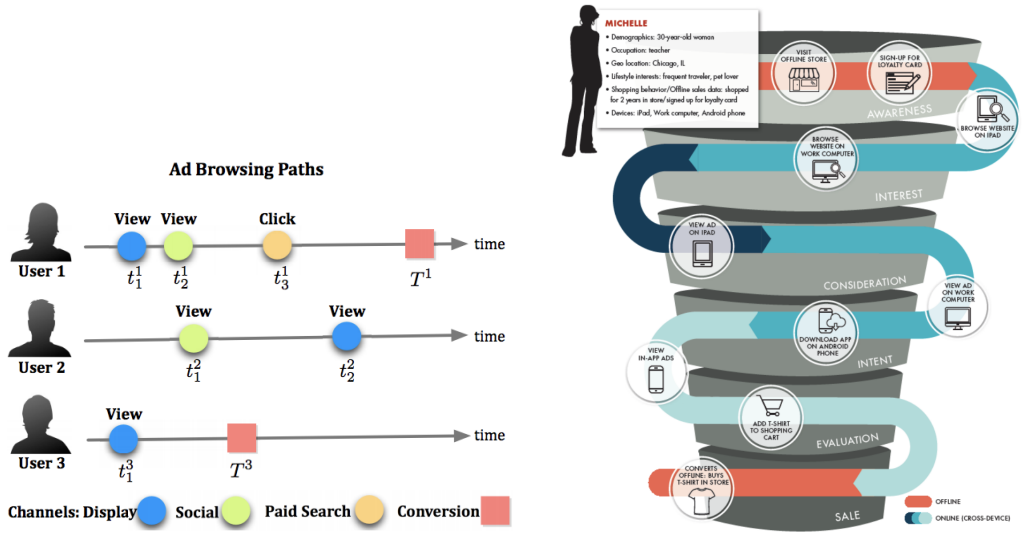

Figure 2.3: Attribution model examples (CIKM) and a user's path to purchase example(10) (IAB UK)

IAB(10) indicates that models such as the Last Click are limited and do not reflect the multi channel scenario we live in today(10). That's why, as Birkbeck(13) explains, studying the effect of different channels in a user's influence is important. Only this way, Birkbeck(13) claims that advertisers will be able to conquer the answers needed to plan and optimize their campaigns.

2.2 Data Mining and Outlier Detection

2.2.1 Outliers

As Hawkins(15) defines, outlier is an observation on a group of data that deviates so much from other observations, enough to arise suspicions that it was generated by a different mechanism. Aggarwal(11) also says that it's a point significantly different from most other points. Ben-gal(16) emphasizes though that outliers can be errors or noise, but they can also be interesting information. (Figure 2.4)

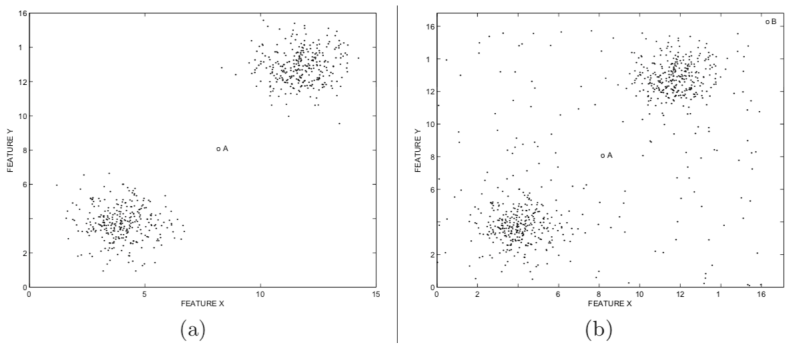

Figure 2.4: On the left, point A is an outlier among two data groups. On the right, the same outlier is next to a lot of noise, but remains as the only real outlier.(11)

Aggarwal(11) explains that outliers can be considered noise or anomalies, depending on its strength. Weak outliers are considered noise and strong outliers are considered anomalies(11).(Figure 2.5)

Ben-gal(16) points out that one of the first steps to get a coherent analysis is detecting outlier observations. Outliers detection, as Steinbach(12) defined, is an analysis that consists in finding observations that are different from most of the data. And this is possible, as Steinbach(12) details, because those anomalous observations have attribute values that deviate considerably from what is expected(12). In his paper, Steinbach notes(12) that even though they're considered rare events, it doesn't mean that outliers do not occur frequently.

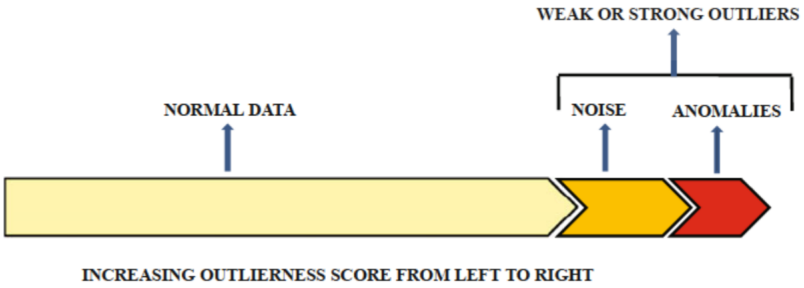

Figure 2.5: Classification of an outlier according to its strength.(11)

In his book, Aggarwal(11) lists systems of industries that most benefit from outlier detection:

(i) Intruders detection systems. In computer systems, data is colected to detect activities that may indicate user's unusual behavior. Those behaviors may indicate malicious activity and are usually labeled as "intruders detection";

(ii) Credit card fraude. A sensitive information such as the credit card number may be easily compromised. The non-authorized user of a credit card may reveal itself through unusual buying patterns, such as high value purchases on different geographical locations;

(iii) Event mapping detection. For some applications, sensors are used to detect sudden changes in patterns that may represent a user's newly discovered interest;

(iv) Medical diagnostic. In a medical investigation situation, data is collected from different sources such as MRIs, PETSCANs or ECGs. Unusual patterns generally reflect some sort of disease;

(v) Police enforcement. There are cases in which uncommon patterns are only possible to be detected with the pass of time, after multiple actions from an entity.;

(vi) Geography. A great amount of spatio-temporal data is necessary to allow detecting climate changes, weather and temperature. These data are usually collected by satellite or remote sensors.

2.2.2 Outlier detection techniques

This sections presents several techniques for outlier detection, which are mostly used in practice.

2.2.2.1 Statistical techniques

Statistical techniques verify if a dataset fits in specific statistical models, as Chandola et al.(17) explain. Data points that have low probability of being generated by the model, based on the use of a statistical tail test, are declared anomalies(17). Those statistical methods can be either parametric or non-parametric. Parametric models assume some distribution parameters(18) and non-parametric models do not assume anything(19).

Parametric methods

As Chandola et al. determine in his survey(17), parametric techniques can be separated either in Gaussianas, Regressions or Mix of Distributions.

Gaussian techniques

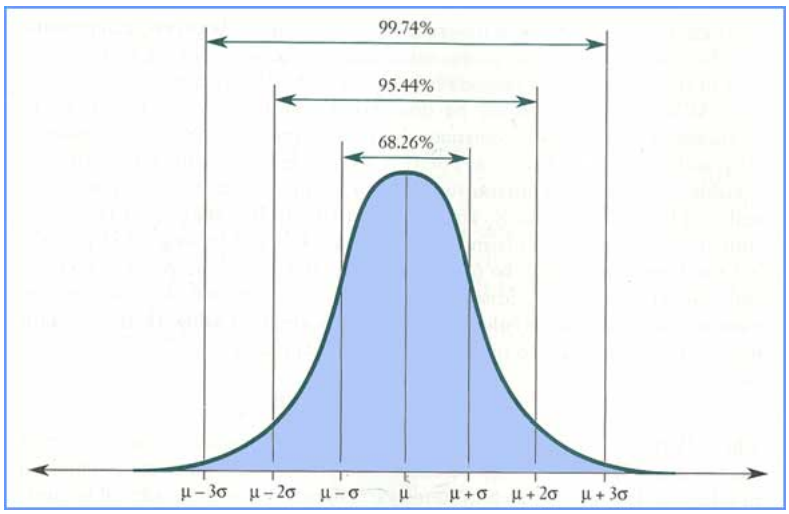

Figure 2.6: Example of a group of data points with a gaussian or normal distribution. 68.26% of the data points are up to one standard deviation away from the mean, 95.44% up to 2 standard deviations away and 99.74% up to 3 standard deviations away. In other words, less than 1% of the data points is more distant than this from the mean. Thus, following the gaussian technique, those points would be considered outliers.(17)

Gaussian techniques (Figure 2.6) assume data points were generated by a gaussian distribution(20). Parameters such as the mean are estimated using Maximum Likelihood Estimates (MLE). Distance from any point to the mean is the anomaly's outlier score. A threshold is applied to determine which points are outliers or not. Chandola et al.(17) say that a simple gaussian technique example is the one created by Shewhart(21), which declares that any point more

than three standard deviations away from the mean is an outlier. More sophisticated tests using gaussian distributions have been discussed by Lewis et al.(22), Barnett(23) and Beckman et al.(24).

The Boxplot technique

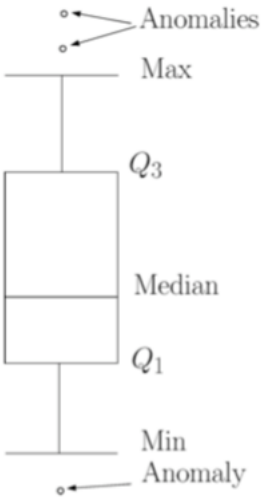

Figure 2.7: Example of a boxplot containing three anomalous points.(17)

Chandola et al.(17) also state that another simple rule applied is the Boxplot(25, 26, 27). A boxplot (Figure 2.7) describes a group of data using atributes as the first quartile (Q_1), median, third quartile(Q_3), minimum non-anomalous value and maximum non-anomalous value. The difference between $Q_3 - Q_1$ is the interquartile range (IQR). Any point that's less than $Q_1 - 1.5 * IQR$ or bigger than $Q_3 + 1.5 * IQR$ is considered an outlier.

Regression techniques

Regression techniques to detect outliers, as explained by Chandola et al.(17), divide themselves in two steps. On the first step, a regression model fits(28) the data. The regression model then tries to create an equation that's able to define all data points available.

For each point (Figure 2.8), the model creates a representation. If this equation is not capable of defining the data points completely, an error variable will be added to it. This error is the residual, the part of the data points that the model couldn't explain. On the second step, the model uses each point's residual as the outlier score. The bigger the difference between the real point and its representation in the model, more it will be considered an outlier. Regression

techniques were used to investigate temporal series data by Abraham et al.(29, 30) and Fox(31).

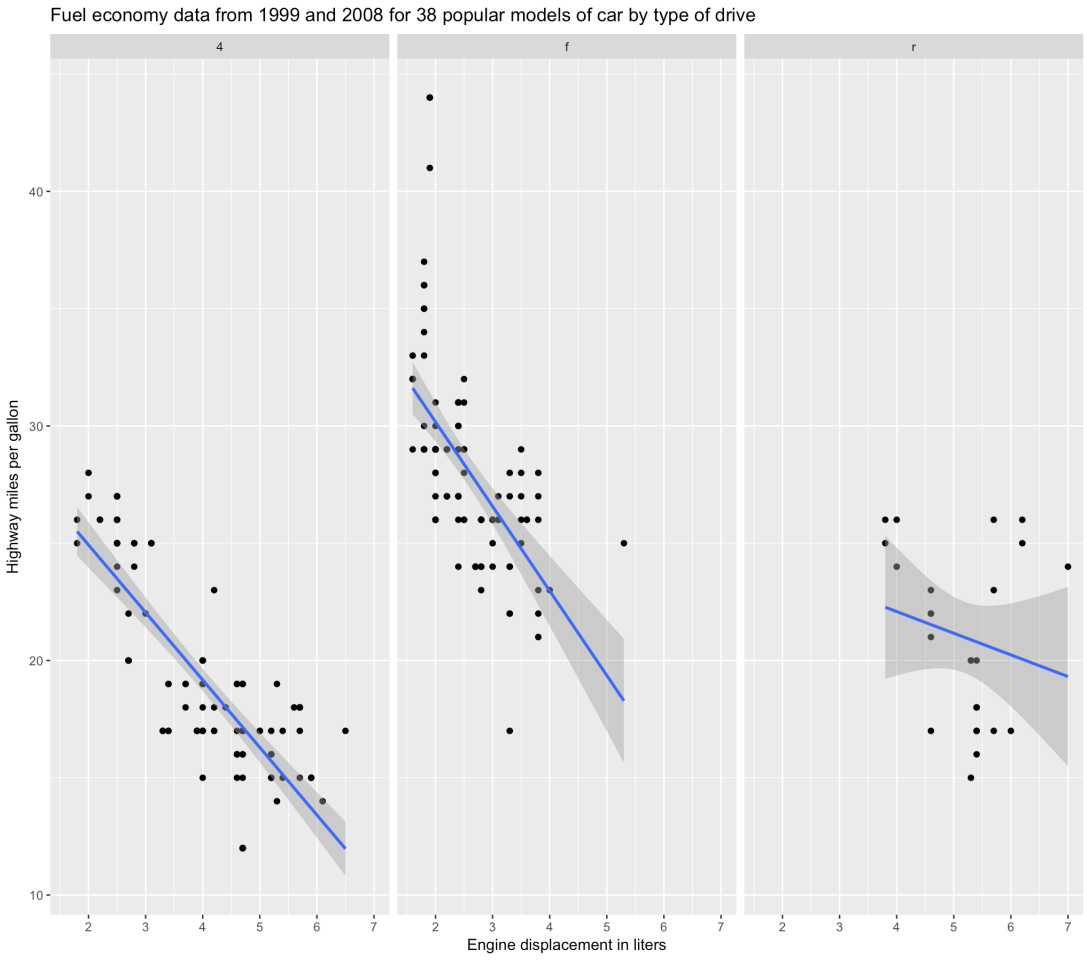

Figure 2.8: Examples of linear regressions. The black dots are part of the group of data points. Crossing the black dots, the blue line represents the linear regression, trying to define the group in a single line. According to this technique, the bigger the distance between the point and the line, more the point will be considered an outlier.(17)

Techniques based in mix of distributions

Chandola et al.(17) conclude that the third category of parametric techniques concentrates methods based in mix of distributions. This category by itself is divided in two groups. First group of techniques, as Abraham et al.(29) explain in his study, assumes that normal data, in other words, data that is not anomalous, were generated by a gaussian distribution. And that anomalous data were also generated by a gaussian distribution, though with a bigger variance. The second group of techniques of mix of distributions, determined by

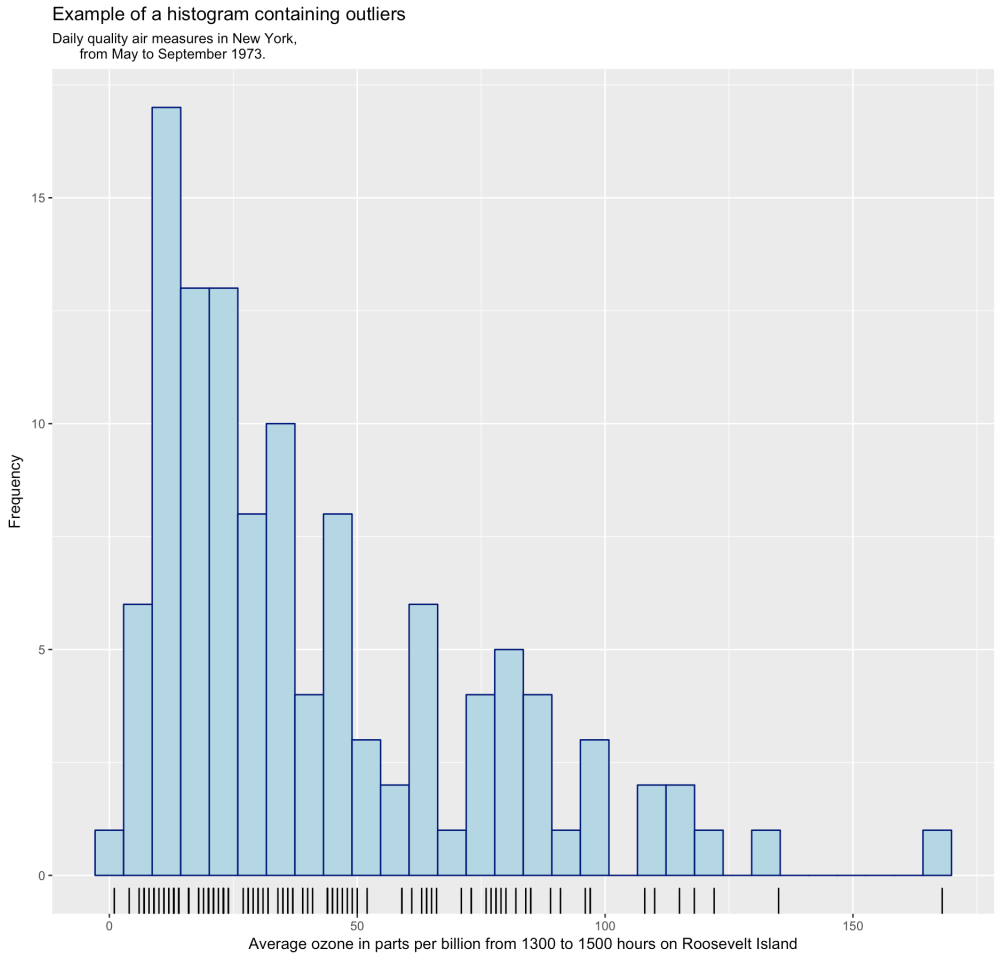

Figure 2.9: Example of a histogram containing outliers. Points located in bins with a height that's too short would be considered anomalies.

Aggarwal(32), assumes normal data were generated by mix of different distributions. And that points that do not belong to any of those distributions are anomalies.

Non-parametric methods

Chandola(17) reveals that non-parametric statistical techniques do not define the structure of the model a priori, but from the data. One of the most simple non-parametric technique(17) is the one that uses a histogram to profile the data (Figure 2.9). Univariate techniques[2] consist in building a frequency histogram based on the unique values of one of the attributes of the dataset, explains Chandola(17). The model defines as anomalies the points that either don't fall in any of the histogram's bins or that fall in bins for which the height is too small.

Histogram techniques require that the data points have a normal distribution, as Anderson et al.(57, 58, 59) discuss in their paper. And the size of the bins

[2]Univariate techniques deal with only one of the variables or dimensions of a dataset.

used to build the histogram is what defines how many outliers the analyst may end up finding. Histograms were also used in multivariate techniques[3]. Those techniques consist in, according to Chandola(17), building a histogram for each of the dimensions in the dataset and calculating the outlier score by attribute, then later aggregating them. Different multivariate methods with histograms were used in intruders detection systems(60), in network intruders detection(61, 62, 63), in fraud detection(64), in structure damage detection(65, 66, 67), in web-based attacks detection(68, 69) and in anomalous topics detection in texts(70).

Non-parametric techniques were also used to estimate density probability curves with kernel functions(71). The method is similar to the parametric techniques previously described, with the difference being the use of density estimation techniques. Desforges(72) proposed a semi-supeervised method using kernels, that consists in building a density probability curve. Points that fall in area with low probability are considered anomalies.

2.2.2.2 Nearest Neighbour

Nearest Neighbours techniques assume that the expected or the normal part of the data points occurs in dense neighbours and that anomalies occur away from its nearest neighbours, as Altman(72) explains in his paper. The neighbourhood creation process consists in calculating the distance (if there are numeric variables[4]) or the similarity coefficient (if there are categorical variables[5]) between the points and verifying those that are distant from its neighbours.

The Nearest Neighbour technique, as Chandola(17) notes, does not scale to a lot of dimensions and does not apply if all data points have outlier scores. The biggest part of the algorithm choose most relevant points to score. Techniques that divide the data points in partitions are linear in space complexity, but exponential in dimension complexity. Sampling techniques try to solve this, but can be unneffective if the sample is too small. As Aggarwal(14) claims in his book, those techniques are non-supervised and do not assume anything regarding the data points distribution. Semi-supervised methods perform better than non-supervised ones(14).

Chandola et al.(17) believe that those techniques can be separated in two main groups:

[3]Multivariate techniques, on the other hand, deal with more than one dimension in the dataset at once.

[4]Numeric variables can be either continuous or discrete numbers(73).

[5]Categorical variables that can take on one of a limited, and usually fixed, number of possible values, assigning each individual or other unit of observation to a particular group or nominal category on the basis of some qualitative property.(74)

(i) Techniques that use distance between points and its k_{th} nearest neighbours as the outlier score;

(ii) Techniques that calculate the relative density of each point to obtain its outlier score.

Techniques that use distance to k_{th} nearest neighbours

Chandola et al.(17) point in their survey that the simplest nearest neighbour technique bases itself on the definition that the outlier score of a data point is its distance to its k_{th} nearest neighbour. K in this case could be any number. If K is equal to 1, the algorithm calculates the distance of a point to it's 1_{st} neighbour. If K is equal to 10, the algorithm calculates the distance of a point to it's 10_{th} neighbour.

Authors such as Eskin et al.(75), Angiulli and Pizzuti(76) and Zhang and Wang(77) calculate the outlier score of a data point as the sum of its distances from its k_{th} nearest neighbors, for example. But a different way to calculate the outlier score of a data point is to count the number of nearest neighbors that are not more than D distance away from this data point itself, as Knorr and Ng reveal in their papers(78, 79, 80, 81). A more sophisticated technique, hypergraph-based, called HOT, was proposed by Wei et al.(82) in which the authors model the categorical values using a hypergraph and measure distance between two data points by analyzing the connectivity of the graph.

Techniques that calculate relative density of each point

Chandola(17) claims that relative density techniques calculate the density of each point's neighbourhood and declares as an outlier points that lie in a low density neighbourhood. If the dataset has too many regions with different density levels, the algorithm can have high processing time. When that's the case, the algorithm calculates the density relative to the distance's density.

Breunig et al.(83, 84) assigns an outlier score to a data point, known as Local Outlier Factor (LOF). For a data point, the LOF score is equal to the average local density[6] of the k_{th} nearest neighbors of a point divided by the local density of the point itself. A normal point would have local density equal or next to local density of the average of its neighbours. Outliers would have high LOFs. (Figure 2.10) This technique has a $O(N^2)$ complexity.

[6]The local density is calculated with the radius of the smallest hypersphere in which the center is the point itself and that contain its k_{th} nearest neighbours. Local density is K divided by the volume of the sphere.

The COF (Connectivity-based Outlier Factor) technique was proposed by Tang et al.(85) to improve the efficiency of the LOF technique. The difference is the way the neighbourhood of K is calculated. Also it uses incrementality to grow the neighbourhood. The first neighbour is the one nearest to the point. The next neighbour is the one which the distance to its neighbourhood is the smallest amongst all the neighbours. This grows until it gets to K.

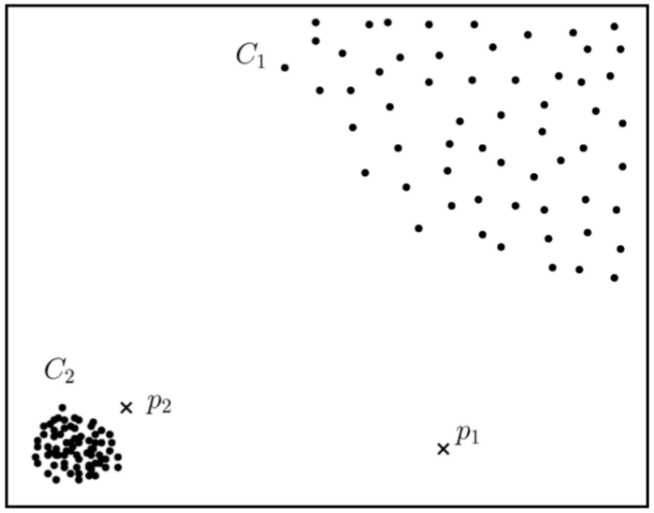

Figure 2.10: Points located near low density neighbourhoods are considered outliers.

Hautamaki et al.(86) proposed a simpler version of the COF method which calculates the outlier score using in-degree number (ODIN - Outlier Detection using In-degree Number) for each data point. According to the technique, the ODIN of a point would be the number of K nearest neighbours which have the data point in particular in their K nearest neaighbour list as well. And the inverse of the ODIN would be the outlier score of the point.

Another relative density technique pointed out by Chandola et al.(17) is the MDEF, which is the Multi-Granularity Deviation Factor. This technique was proposed by Papadimitriou et al.(87). The MDEF of a point would be the standard deviation of the local densities of the nearest neighbours of this point (including the point itself). The inverse of the MDEF would be the outlier score. This technique can also find anomalous micro-clusters, Chandola points out(17).

2.2.2.3 Clustering

In his survey, Chandola reveals(17) that clustering techniques, though initially used to find clusters, are also used to find outliers and anomalies. One thing the author points out is the fact that those techniques could have quadratic

time complexity if they have to calculate parwise distance for each of the data points. The pros, Chandola(17) cites, are that those can be adapted to complex data and the cons are that their efficacy depend on their capacity on capturing a coherent cluster structure. These techniques can be divided into three groups.

Techniques that assume that inlier points belong to some or any cluster

The first group of clustering outlier detection techniques assumes normal points belong to any cluster in the data, while outliers do not. So the clustering algorithm partitions the points into clusters. And any point that does not fall into any cluster is considered an outlier. In this category, a clustering algorithm will only be fit for detecting outliers if it doesn't force a point to be part of a cluster (like K-means would do[7]). Ester et al.(88) proposed DBSCAN, a density-based clustering algorithm that groups together points that are closely packed together (Figure 2.11); Guha et al.(89) proposed ROCK, a robust clustering algorithm for categorical attributes; and Ertoz et al.(90) proposed SNN, an algorithm that finds clusters of different shapes, sizes and densities in high-dimensional data - those are all algorithms that can be used.

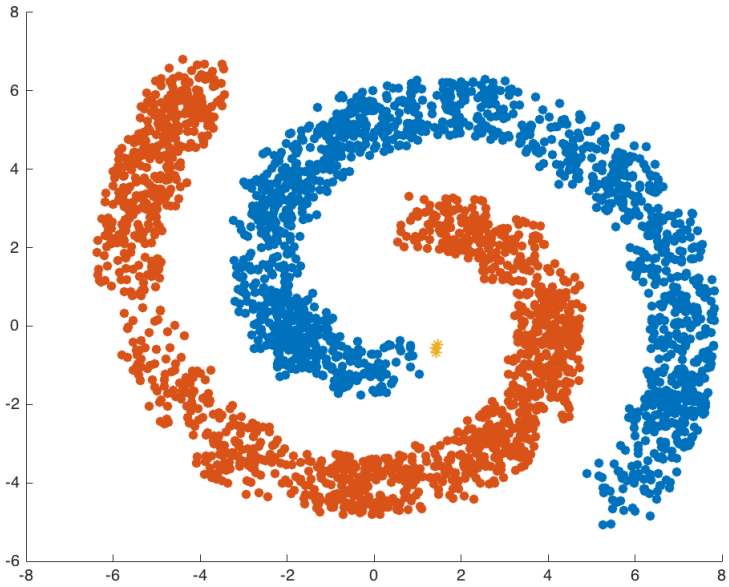

Figure 2.11: Example of DBSCAN outlier detection algorithm dividing the data points in two clusters.

Another algorithm that falls into the first category is the FindOut - proposed by Yu et al.(91), an extension of the WaveCluster algorithm - proposed by

[7]K-means is an algorithm described by Lloyd in 1957 that partitions n points of data into K clusters in which each point belongs to the cluster with the nearest mean.

Sheikholeslami et al.(92), as explained by Chandola(17). This technique detects clusters and removes them from the data. Then, any residual points that remain are declared outliers.

Techniques that assume that inlier points lie close to their cluster centroids

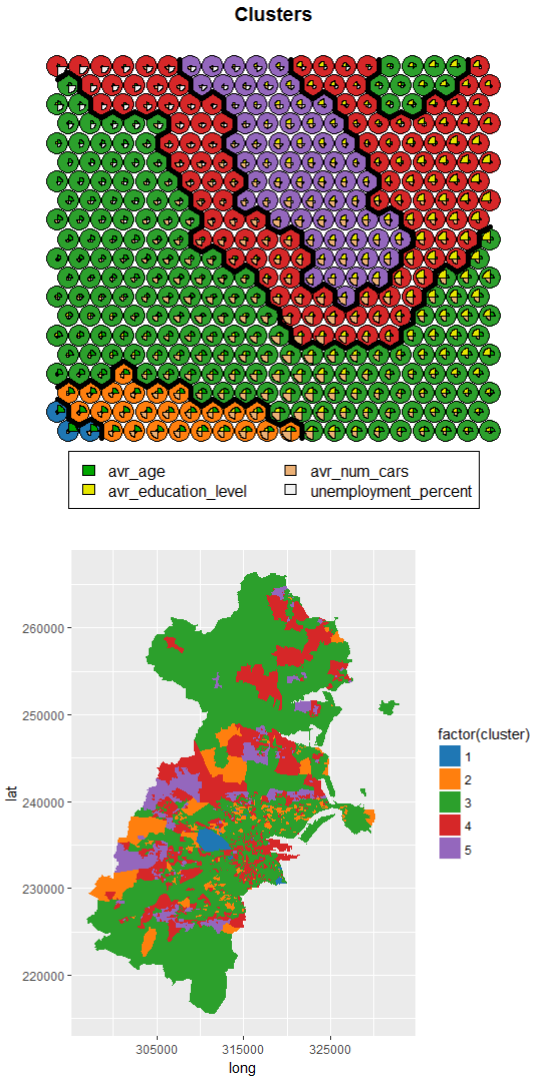

Figure 2.12: The inhabitants of Ireland clustered by demographics with the SOM algorithm. And the Ireland map colored by those same clusters.

The second group of clustering outlier detection techniques assumes normal points lie close to their cluster centrois, while outliers are far away from it. They consist each in two steps, as Chandola(17) notes: first, the data is clustered, then, for each point, its distance to its closest cluster centroid is considered its outlier score. Techniques such as the Self-Organizing Maps (Figure 2.12), K-Means and the Expectation Maximization (EM), studied by Smith et al.(93) can

be used to cluster and then classify outliers. Those techniques can also operate on supervised mode, as noted by Chandola(17), in which the training data is clustered and points belonging to the test data are compared against the clusters to obtain an outlier score for the test data point. The cons of techniques like these are that if the outliers in the data are able to form a cluster themselves, then the algorithms won't consider any point as an outlier.

Techniques that assume that inlier points belong to large and dense clusters

The last group of clustering outlier detection techniques assumes that normal data points belong to large and dense clusters while outliers belong to small or sparse clusters. Chandola(17) states that those techniques declare that points belonging to clusters whose size and/or density are below a threshold as outliers. Several techniques have already been proposed(94, 95, 96, 97, 98, 99).

The work proposed by He et al.(99), FindCBLOF, assigns an outlier score - the CBLOF (Cluster-Based Local Outlier Factor) - for each point. This score captures the size of the cluster to which the point belongs to, as well as the distance of the point itself to its cluster centroid. And the point will only join a cluster which centroid distance to it is smaller than the threshold created. If this cluster does not exist, then a new cluster with this point is created. And the algorithm determines which clusters are anomalies based on density and distance to other clusters. Some works use this technique using K-D-Trees[8] partitioning the data in linear time.

2.2.3 Facts that impact on the quality of an outlier detection algorithm

Outlier detection algorithms require, as Aggarwal(14) argues, careful criteria to be relatively compared to one another. And outlier detection is usually more difficult to analyze because:

(i) Sample space is usually small, which makes it difficult to verify assertiveness of an algorithm with robust statistics;

(ii) There's no ground-truth, which means outlier detection problems are mostly unsupervised problems; they lack a set of true examples that can be used to calibrate the algorithm;

[8] A k-d tree (short for k-dimensional tree) is a space-partitioning data structure for organizing points in a k-dimensional space.(100) A k-d tree can help find the nearest neighbour to a specific point in space without the need to explore all the partitions.

(iii) Context can be a problem; contextual outliers are even harder to detect because their interpretation depends on the problem being analyzed.

2.2.3.1 The Ground-truth and the labels

Aggarwal(11) discusses in his book that opposite to what happens with classification problems, outliers are hard to evaluate, because they are rare. What the author implies is that in the case of detecting outliers, there is not a ground-truth.

The ground-truth is a concept used in supervised problems(101), or problems where a learning function can map inputs to outputs based on example input-output pairs. Those examples are the ground-truth, which are used as parameters to tune the algorithm towards being better at classifying the data points. Aggarwal(11) reminds us that in the case of unsupervised problems, there is no ground-truth, which means the function cannot be tuned by any examples, because they do not exist. And since the function cannot know if it's correctly learning the problem or not, it cannot be quantitatively evaluated, the author concludes(11).

So what happens in case of unsupervised outlier detection problems, Aggarwal reveals(11), is that researchers use real use case studies in order to provide intuitive and qualitative evaluation of the outliers. In other words, real outliers are used as examples for the ground-truth while the unsupervised algorithm detects any new underlying outliers in new datasets.

And in order to create a ground-truth for the algorithms is to provide labels. Aggarwal debates(11) this is a process that can be slow and tiresome, because mostly it's a manual work. Real use case examples of outliers are labeled as such and then provided as parameter to the algorithm in order to help measure Precision and Recall effectiveness.

2.2.3.2 Context

Another fact specific to the outlier detection problem that heavily impacts on its quality is the context. Chandola(17) explains there exists three types of outliers:

(i) Outlier points. When a single point or a collection of individual points are considered outliers;

(ii) Contextual outliers.

  (a) When you have to consider the context in order to define an outlier.;

(b) Each point has always two types of attributes: contextual (determine the neighbourhood; e.g. latitud and longitud for spatial data and time for time data) and behavioral (general characteristics);

(c) When we say the anomalous behaviour is coming from a specific context, we call this a contextual outlier. E.g. In a dataset of time series data, a 35ºC temperature might be normal in general, but in the context that this characteristic is happening during winter, this temperature becomes an anomaly. The degree is a behavioural attribute, the month is a contextual attribute;

(iii) Collective outliers.

(a) When the group of data is an anomaly related to the whole dataset;

(b) Points individually may not be considered an outlier, but if happening together they are. E.g. A signal drop in a ECG is not abnormal, but if happening for a long period of time and lots of times it may be troublesome (Figure 2.13);

(c) It only happens when the points have a relationship among themselves.

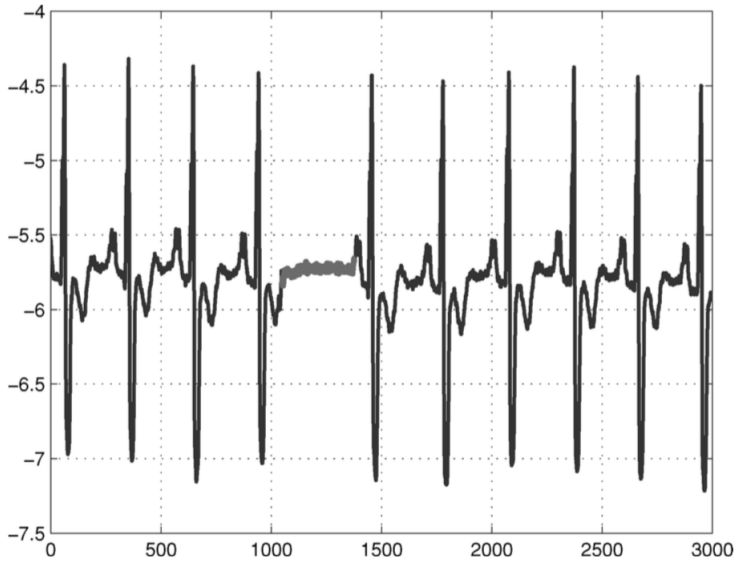

Figure 2.13: Example of an ECG report where a drop of signal happening for a long time could indicate a heart failure.

2.3 Ensemble Learning

Ensemble Learning, as Zhihua(33) explains in his book, is a machine learning method that uses multiple algorithms to obtain better prediction results than individual algorithms would do otherwise. Ensembles tend to be used in data mining problems, according to Aggarwal(14), because they reduce dependency that a single algorithm can generate.

The author(14) defines a few reasons on why to use ensembles and not individual outlier detection algorithms to find anomalies:

(i) Ensemble for outlier detection was used in a limited way. Instead of proposing real ensembles, Aggarwal argues other authors have proposed individual algorithms that themselves benefit from other methods, creating what the author calls pseudo-ensembles. In other ways, most techniques proposed are not formally described as ensembles and thus do not discuss the theoretical challenges as such;

(ii) Using outlier detection ensembles reduces the dependency creating from individual algorithms, increasing robustness of the process.

These are the components that compose the creation of an ensemble in a very generic way:

(i) Creation of a model or mixture of individual models;

(ii) Normalization of scores. Different methods will create scores in different scales. A normalization is needed;

(iii) Model combination. The final combination function decides on which algorithms to use and when.

This meta-algorithm creation method was used in contexts such as clustering, classification and outlier detection. In clustering, ensemble has been applied to multiview clustering and to alternative clustering, as Aggarwal(14) states. The ideia is that each one of the variables behind a clustering analysis is subjective and therefore do not completely reflect data. Thus, one would have to examine different and alternative clusters(36, 37, 38, 39) to combine their results. Alternative clusters are also called multiview because, as explained by Aggarwal(14), they can be visually analyzed as in the works of Hinneburg et al.(40, 41) to obtain different insights. To demonstrate how cluster ensembles can be similar to outlier detection ensembles, Moosmann et at.(42) compare in his text the clustering ensemble Extremely-Randomized Clustering Forest (ERC) to the outlier detection ensemble Isolation Forests(43).

In the context of classification, a variety of methods based in ensembles have been proposed such as Bagging(44), Boosting(45), Stacking(46, 47, 48), Random Forests(49, 50), Model Averaging(51) and Bucket of Models(52). Ensembles usage is particularly important for classification, as Aggarwal discusses in his book(14), when the quality of results for individual classifiers isn't robust enough due to limitations of data's own quality or of time of processing. Most recent methods include outlier detection techniques for unbalanced classes, as in the works of Micenkova et al.(53, 54, 55, 56).

2.3.1 Outlier Ensembles

One of the main reasons why ensembles should be used in outlier detection, as Aggarwal(14) argues, is the fact that most outliers are found not in univariate data but in multivariate data, in other words, in datasets with medium or high dimensionality. The author(14) explains that to be able to account for collectivity and contextuality, one would have to look in many subspaces within data's dimensions in order to locate subsets of points that would be considered outliers. What happens when individual algorithms are used is that "only one subset or a small group of subsets of data are analyzed, making any outlier finding a guess". The use of multiple models reduces the incertaintity of the subspace selection inherently hard and guarantees more robustness to the method. Different methods have been proposed to account for high-dimensional outlier detection such as(103, 104, 105, 106, 107, 108).

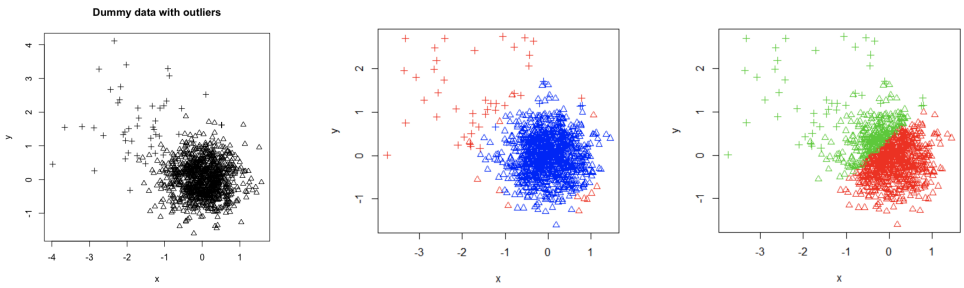

Figure 2.14: Example of Outlier Detection Ensemble iForest declaring outliers. To the left, I introduced two random distributions of 1000 points centered around mean 0 with a standard deviation of 0.5. Then added 50 outliers centered around -1.5 and 1.5 with a standard deviation of 1. Records that exceed the 95% percentile of the anomaly score flag the most anomalous records and are colored red. To the right, an example of K-Means declaring outliers in the same dataset. Since K-Means declares as part of the same cluster all points around the centroid, it's not capable of declaring outliers in a depth basis.

The feature bagging proposed by Lazarevic et al.(105) may be considered a first formal description of outlier ensemble analysis in a real setting. Another

example is the isolation forest (Figure 2.14) proposed by Liu et al.(43), which quantifies the dimensionality of the local subspace in which a point can be isolated as an outlier after using random splits in a decision tree-like fashion. According to the author, outliers can be isolated in low-dimensional subspaces when randomly partitioning the data space. Another method known as rotated bagging, proposed by Aggarwal et al.(109), generalizes the feature bagging ideas to subspaces defined by combinations of arbitrarily oriented vectors and not just axis-parallel combinations of features.

2.3.2 Methods for evaluating ensemble performance

2.3.2.1 The Precision-Recall Curve

Outlier detection algorithms output a score and a threshold is used in order to declare points as outliers. Though this sounds like a solution, it puts yet another challenge. Because if the threshold is picked too restrictively, it will minimize the number of outliers, but can also miss true outliers (or bring a lot of false negatives). On the other hand, if the threshold is picked too widely, it will declare too many points as outliers and maybe lead to many false positives. This trade-off can be measured in terms of Precision and Recall.

Since Precision and Recall define true positives and false positives for classification problems, in order to adapt those concepts to the outlier detection scenario, Aggarwal(11) suggests a slightly different formula for both Precision and Recall.

Let's say we declare an outlier set S(t). Let's also say that G represents the true set of outliers (the ground-truth, used from real use cases). Finally, let's say t would be the threshold we choose to declare outliers in a given algorithm. S(t) changes as t changes, because as mentioned before (11) the threshold impacts on the number of outliers that will be declared. Then the Precision is a measure that defines the percentage of outliers that truly turn out to be outliers.

$$Precision(t) = 100 * \frac{|S(t) \cap G|}{|S(t)|}$$

Recall on the other hand is defined as the percentage of ground-truth outliers that have been declared as outliers by the algorithm with the given t (the opposite of Precision).

$$Recall(t) = 100 * \frac{|S(t) \cap G|}{|G|}$$

By varying t, it's possible to measure the trade-off between a high precision or a high recall, by plotting the Precision-Recall curve (Figure 2.15). For more effective algorithms, high values of Precision may often correspond to low values of Recall and vice-versa (returning a high percentage of true positives and a low percentage of false positives).

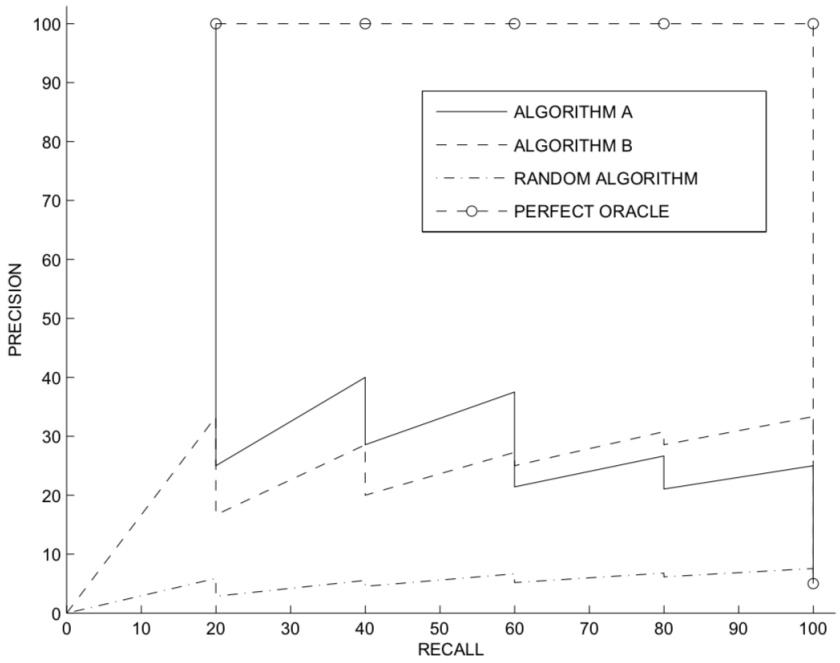

Figure 2.15: Precision-Recall curve of 4 algorithms plotted for comparison. The ground-truth (perfect oracle) would have always 100% Precision while Recall would depend on the threshold used. For other algorithms, this would depend on their effectiveness. Source: (11)

2.3.2.2 The Receiver Operating Characteristics Curve

The ROC Curve (Receiver Operating Characteristics Curve) is similar to the Precision-Recall, only as Aggarwal points out(11), visually more intuitively. Instead of using Precision and Recall, in the ROC curve TPR (True Positive Rate) and FPR (False Positive Rate) are graphed against each other. The TPR is defined in the same way as the Recall. And the FPR is the percentage of the points that were reported as outliers, but were not true outliers, out of total ground-truth outliers.

So, let's say we define D for the ground-truth points that are not outliers, the measures are defined as:

$$TPR(t) = Recall(t)$$

$$FPR(t) = 100 * \frac{|S(t) - G|}{|D - G|}$$

The end points of the ROC curve are always at (0,0) and (100,100). As Aggarwal(11) notes, a random algorithm is expected to show performance along the diagonal line connecting these points. The lift obtained above this diagonal line could be used to consider that an algorithm is more accurate than a random one. The discovery of a new outlier at any particular relaxation in rank threshold results in a spike in the precision, which becomes less pronounced at higher values of the recall.

2.3.3 Ensembles categorization

The position paper proposed by Aggarwal(102) categorizes outlier ensembles in three different ways:

(i) By component independency:

  (a) When the execution of an individual component either depends on the result of previous run components or it doesn't.

(ii) By centrality of the ensemble:

  (a) Each component either centralizes around data or the model;

  (b) In other words, either we're subsampling a part of the data or we're choosing an algorithm;

  (c) Data-centric ensembles are usually independent and model-centric ensembles are usually sequential.

(iii) By theoretical approach:

  (a) Outlier detection ensembles are always either trying to reduce bias or variance. Whenever reducing too much variance, they create bias and vice-versa. The trade-off has been greatly discussed by Aggarwal(14) in his book.

  (b) Bias reduction is always more difficult in the case of outlier detection because of the absence of the groumd-truth.

Table 2.1 separates recently proposed works in outlier detection ensembles by categorization.

Table 2.1: Table example with some ensemble techniques categorized by centrality and dependency

Method	**Model-centered or Data-centered**	**Sequential or Independent**	**Combin. function**	**Normaliz.**
Feature bagging	Data	Independent	Max/Avg	No
HiCS	Data	Independent	Selective Avg.	No
LOF tuning	Model	Independent	Max	No
LOCI tuning	Model	Independent	Max	No
Isolation forests	Model	Independent	Average	Yes
Multiple-Proclus	Model	Independent	Harmonic Mean	No
RS-Hash	Both	Independent	Average	No
OutRank	Model	Independent	Harmonic Mean	No
Calibrated bagging	Both	Independent	Max/Avg	Yes
SELECT	Model	Sequential	Average	Yes
OUTRES	Data	Sequential	Product	No
Rotated bagging	Data	Independent	Max/Avg	Yes
Converting scores to probabilities	Both	Independent	Max/Avg	No
Intrusion bootstrap	Model	Independent	Last Component	No
Nguyen et al.	Both	Independent	Weighted Avg.	No
Entry subsampling (Graph Matrix)	Both	Independent	Median	Yes
Bagging	Data	Independent	Average	Yes
Variable subsampling	Data	Independent	Many	Yes
One-class	Data	Independent	Avg. Product	No
LODA	Model	Independent	Average	Yes
CARE	Both	Sequential	Weighted Avg.	Yes
FVPS	Data	Sequential	Average	Yes
RandNet	Model	Independent	Median	Yes
BSS/DBSS	Data	Independent	Average	Yes

2.3.3.1 By component independency

Dependent ensembles: sequential

In sequential ensembles, Aggarwal(102) states, the set of algorithms used are applied sequentially to either a portion of the data or to all of it. The ideia is that one application is impacted by others that were applied before. The author explains this impact means either the previous algorithm applied modified the data or prevented another specific algorithm from being applied. The final result is either a weighted combination of all outlier scores or the final result provided by the last algorithm used. The method can run a fixed number of times or run until it converges to some result.

Algorithm 1 Sequential Ensemble

```
1: procedure ENSEMBLE PROCESS(Dataset, Alg_1...Alg_n)
2:     j ← 1                                                    ▷
3:     for j ← 1...n do                                         ▷
4:         Alg_n ← randomAlgorithmBasedOnPastExecutions          ▷
5:         f_n ← newDatasetFromDbasedOnPastExecutions            ▷
6:         f_j ← Alg_n                                           ▷
7:         j ← j + 1                                             ▷
8:     outliers ← combinationOfPreviousExecutions                ▷
```

Aggarwal(102) argues though that sequential ensembles have been used less for outlier detection than independent ensembles. Some works propose to remove initial outliers from the data in order to get more robust results from other algorithms. The refinement the data suffers along sequencing of the algorithms could be either: data subset selection, attribute-subset selection or generic data transformation methods.

Aggarwal(102) shares an example algorithm (Algorithm 1) to show how sequential ensemble works. Let's say we have a 2-phase sequential ensemble where in the first phase it uses a base detector to remove more obvious anomalies. And in the second phase, it uses a more robust method to search for outliers in a new dataset already without noise. An ensemble like this was proposed by Barbara et al.(110). Another example could be an ensemble where the models recursively look for outliers in subspaces of the data, but only in different subspaces where its previous predecessors algorithms haven't looked yet. Muller et al.(111) proposed a work similar to this one.

Rayanaand and Akoglu(112) proposed a model-centric sequential ensemble for detecting outliers that first executes base detectors to create a pseudo-ground-truth by averaging the scores of these detectors. The artificial ground-

truth is then used to prune detectors that deviate from average performance. This works to remove poorly performing detectors and improve overall performance. Final scoring is an average of the independent detectors, but the selection step requires sequential evaluation and therefore it is considered a sequential ensemble.

Independent ensembles

On the other hand, Aggarwal(102) explains, in the case of independent ensembles, instead of sequencing algorithms that depend on one another to be executed, different instantiations of the same algorithm can be used on different portions of the data. The same algorithm may be applied with either different initializations, parameter sets or random seeds. Results can be combined in order to obtain a more robust outlier score.

Independent ensembles are the most common for the outlier detection problem. Aggarwal(14) provides an pseudo-algorithm example (Algorithm 2) of the independent ensemble. The ideia behind using independent ensembles is that different models (or different combinations of the same models) can provide different and valid insights about data. Combining these insights could provide more robust results which are not dependent on specific artifacts or any algorithm or dataset.

Aggarwal(14) states that this method has been particularly used for high-dimensional data, since it enables the exploration of different subspaces of the data where different deviants may be found. Basically, the author explains, independent ensembles can be used in any setting with a randomized base component in which we expect predictions to vary from one component to another.

Algorithm 2 Independent Ensemble

```
1: procedure ENSEMBLE PROCESS(Dataset, Alg_1...Alg_n)
2:     j ← 1                                                    ▷
3:     for j ← 1...n do                                         ▷
4:         f_n ← newDatasetFromDbasedOnPastExecutions           ▷
5:         f_j ← Alg_n                                          ▷
6:         j ← j + 1                                            ▷
7:     outliers ← combinationOfPreviousExecutions               ▷
```

Some independent ensemble methods have been proposed such as (113, 105, 43). They bag the features in sampled subsets of the data in order to get more robust results. Aggarwal(14) says that some classification methods such as(109, 114) which use bagging and subsampling can also be "trivially" adapted to outlier ensembles.

2.3.3.2 By centrality

Typically, it is possible to create an ensemble where both different portions of the data are explored and different models are also used to get more robust results. Aggarwal highlights(14) that so far only Nyugen et al.(115) proposed such a thing. Most other outlier detection ensembles proposed either to work on different portions of the data or to use different models in all the dataset itself.

Model-centric ensembles

Model-centric ensembles combine different models or algorithms built on the same dataset. And Aggarwal(14) points out that one of the major challenges when building model-centric ensembles is that scores from different models are not necessarily comparable to one another, for example the outlier score for the K_{th} Nearest Neighbour approach and the PCA (Principal Component Analysis) approach generate very different outlier scores because they're on different scales. Sometimes larger scores may indicate a stronger outlier, whereas in other cases smaller scores may do the same. This means that when combining the scores, one has to be able to convert and normalize values to make them interpretable. The work proposed by Gao and Tan(116) does this by converting scores to probability.

Another challenge, as Aggarwal(14) explains, is to choose the combination function for the outlier detection: between model averaging, best fit, worst fit and so many other choices. The LOF method proposed by Breunig et al.(83, 84) runs the model over a range of values of K, which is the neighbourhood of the points. On the other hand, the LOCI method proposed by Papadimitriou et al.(87) samples the neighbourhood in order to determine the level of granularity in which to compute the outlier score. Different sampling neighbourhoods are used to detect which algorithm enhances the outlier behaviour of that point the most.

Model-centric ensembles can also use a single algorithm but randomize its base detector. The work proposed by Chen et al.(117) uses randomized autoencoder ensembles. The authors use the autoencoders to perform nonlinear dimensionality reduction and then use the average of the residuals from the outputs to create the outlier scores. OutRank, proposed by Nguyen et al.(108), combines results of multiple rankings based on the relationship of the points to their nearest subspace clusters, uses the aggregate of the normalized fraction of points and the normalized fraction of dimensions in a subspace cluster in order to score all points present in it. For when clusters overlap, Aggarwal et al.(118) proposed another technique, PROCLUST. The idea is to score each data point

based on its relationship to its nearest cluster as well as the properties of its nearest cluster. The score of a point in a single ensemble component is obtained by summing this value over all clusters in which the point lies. The averaged score over multiple randomized executions is used to quantify the outlier score of the point.

Data-centric ensembles

Data-centric ensembles explore different parts, samples, projections or functions of the data instead of different models. The idea behind this approach, as Aggarwal points out(102), is that each part of the data could provide different insights. Feature bagging(105) is a type of data-centric ensemble used for outlier detection. Below, the pseudo-algorithm (Algorithm 3) gives an idea of how it works. Random subspaces of the data are sampled, outlier scores of the data points are calculated in these projected subspaces. Final outliers are a combination function of different scores from different subspaces.

Algorithm 3 Feature Bagging

```
1: procedure ENSEMBLE PROCESS(D)
2:     for j ← 1...n do                                          ▷
3:         integerR ← sample(d/2, d − 1)                         ▷
4:         rDimensionalProjection = select(Rdimensions)          ▷
5:         LOF_Score ← pointsINprojection                        ▷
6:         combined_scores ← scoresDifferentSubSpaces            ▷
7:     outliers = combinedScores                                 ▷
```

The feature bagging method uses axis-parallel projections. Rotated bagging(109), on the other hand, generalizes this method to use arbitrary random projection. The method selects a randomly rotated axis-system of low dimensionality (which is $2 + *\frac{\sqrt{d}}{2}$). However, the axis system is randomly rotated and data points are projected on this rotated system. This rotation provides better diversity and also discovers subspace outliers in arbitrarily oriented views of the data. The work proposed by Muller et al.(111) develops a novel for outlier ranking using the product of the outlier scores in different discriminative subspaces.

2.3.3.3 By theoretical approach

Aggarwal(102) explains that categorizing outlier detection ensembles by theoretical approach is the same as trying to define how these ensembles will try to reduce The Overall Error. The Overall Error of an outlier detection algorithm

can be decomposed into Bias[9] and Variance[10]. This can be defined by the Mean Squared Error equation.

The thing is, Aggarwal(102) points out that it is not possible to calculate the bias without the ground-truth. And there's no ground-truth available in unsupervised problems as was mentioned before, as is the case of outlier detection problems. To be able to theoretically evaluate the ensembles, one can assume a hypothetical (but unknown) ideal ground-truth does exist. This ground-truth could be either defined in terms of outlier scores or in terms of underlying binary labels.

$$MSE = Bias^2 + Variance$$

In the above equation, the squared bias represents the difference between the output we expect the algorithm will have and the ideal outlier scores (which we do not know because it is an unsupervised problem). Even though, it can still be defined as a theoretical quantity for comparison purposes. The variance represents the mean-squared deviation in the outlier scores over various randomized samples of the base data or randomized trials of the base detectors. Thus, the variance is a result either of the nature of the dataset or of the variations in the output.

Ensembles that reduce the variance

Outlier detection ensemble methods that reduce variance do so to try and improve the robustness of the base detectors used. This is done by using different randomized samples of the data or different randomized trials of the base detectors and then averaging results. The idea, according to Aggarwal (102), is to provide more similar results over different datasets.

Since the variance is part of the overall error equation, reducing it will reduce the error of the detector. Majority of outlier detection ensembles are focused on reducing variance.

Ensembles that reduce the bias

Methods that focus in reducing bias are trying to improve the accuracy of

[9]A statistics is biased when it is calculated in such a way that it is systematically different from the population parameter from which it was estimated from(119).

[10]Informally, variance measures how far a set of (random) numbers are spread out from their average value(120).

the outlier detector in expectation. Cons of theoretically approaching the outlier detection by the bias reduction is that in order to reduce bias, the ground-truth has to be available to guide the algorithm. And the ground-truth is usually not available, as Aggarwal(102) notes, in outlier detection. So one has to use heuristic methods to reduce bias.

2.3.4 Theory of Outlier Detection Ensembles

Defining the bias-variance trade-off

Bias-variance theory decomposes these randomized errors into two parts, each of which can be reduced with a specific type of ensemble-centric design. the model bias defines the basic "correctness" of a model. They can be quantified as a theoretical construct (with respect to the unobserved ground-truth) but it cannot be evaluated in practice for a particular application.

To understand the theory behind the bias-variance trade-off, let's consider a sampled data point $\bar{X}_i$, for which the outlier score was modeled with a training data D. As Aggarwal(102) suggests, for theoretical purposes, one can assume an ideal outlier score y_i exists for this point, even though we don't know what it is. This ideal score is output by an ideal function $f(\bar{X}_i)$ with 0 mean and variance of 1.

$$y_i = f(\bar{X}_i)$$

By applying this standard normal distribution to y_i, we get the relative outlier rank of $\bar{X}_i$ with respect to all possible points generated by the whole data distribution. This function is mapping the score y_i to its percentile outlier rank between 0 and 1. But, as Aggarwal(102) explains, in practice algorithms rarely output scores satisfying this property, so this function would be like an oracle that cannot be computed in practice. And for unsupervised problems, we wouldn't even have the examples to verify the oracle.

The score y_i is analogue to the numeric dependent variable in a regression equation, when we add an additional term to the right-hand side of the formula to denote the error or the noise. In unsupervised problems, Aggarwal(14) notes, there's no need to add any noise, since the dependent variable does not exist in practice. Therefore, since the ideal function for the outlier score of a data point is unknown, we have to estimate it with an outlier detection model $g(\bar{X}_i, D)$. For example, the K_th Nearest Neighbour algorithm would be represented by the below equation.

$$g(\bar{X}_i, D) = \alpha KNN - distance(\bar{X}_i, D) + \beta$$

Here, α and β are constants to standardize the scores to 0 mean and variance of 1. When the estimated function $g(\bar{X}_i, D)$ does not model correctly the ideal function $f(\bar{X}_i)$, the we have errors. This is the model bias, because errors would be sistematically being caused by a parameter. Furthermore, since the outlier score y_i depends on the dataset D, which is finite, even if the *expected* value of $g(\bar{X}_i, D)$ correctly reflects $f(\bar{X}_i)$, this estimation with limited data is unlikely to be 100% correct. So the more different $g(\bar{X}_i, D)$ is of $E[g(\bar{X}_i, D)]$ over random choices of training sets of D, the higher the variance we get. In other words, the variance would be the *inconsistent behavior* the algorithm presents in which the same point receives very different scores over different samples or sets of the training data D. This can happen when the algorithm tries to adjust too much to specific nuances of the training set (in order to reduce the bias), which is called overfitting.

Quantifying the bias-variance trade-off

Quantifying the bias-variance trade-off for outlier analysis using ROC curves is a challenge, because the uniqueness of the score-based output is missing. In other words, outlier scores are relative, as Chandola(17) points out. What it means is that if all scores are multiplied by the same positive quantity or translated by the same amount, various metrics of the outlier detector (e.g., ROC curves) are kept unchanged, because those depend only on the ranks of the scores. So ROC curves provide only an incomplete interpretation of the scores (in terms of relative ranks).

The MSE of the detectors of the test points over a particular instantiation of the training set D could be defined as the below equation.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\{y_i - g(\bar{X}_i, D)\}^2$$

The *expected* MSE over random instantiations of the training data using a random process would be the below equation.

$$E[MSE] = \frac{1}{n}\sum_{i=1}^{n} E[\{y_i - g(\bar{X}_i, D)\}^2]$$

Chandola(17) states that different interpretations of the bias-variance

trade-off will yield different decompositions of the MSE. The traditional view would assume the finite data can only be fully used once and that limitation could cause extreme variability in results. Another interpretation would say that different samples of the same dataset can enable the MSE to be computed several times and thus reduce variability. A third interpretation could say that randomizing the base detector itself would turn process variability towards the model and not the data. Aggarwal(14) emphasizes that it is important to specify the underlying process of decomposition of the MSE to properly analyze the effectiveness of the ensemble method.

That is the reason Aggarwal(102) believes analysts should demarcate training and test data even though it's an unsupervised method. The reason it's because it allows to evaluate effects of randomizing the training data over the same set of points. Then if randomized predictions over the same set of points vary significantly even though the point has an accurate expectation, we can say the model has high variance. On the other hand, if the expected prediction of the point is inaccurate, we say the model has high bias. The idea is to realize which of the two components (variance or bias) need to be improved to reduce overall error.

2.3.5 Quality factors that impact the outlier detection ensemble

2.3.5.1 Scores normalization

Since different algorithms can output outlier scores in different scales, normalization becomes an important step in creating outlier detection ensembles, otherwise scores cannot be compared. In some cases, for example, high outlier scores can correspond to larger outlier tendency whereas in other cases can correspond to low tendency.

Aggarwal(102) calls to the fact that the analyst that creates an ensemble has to be careful not to weight one algorithm more than the other. Furthermore, combining algorithms with different conventions on ordering of the scores can lead to completely unpredictable results. One approach can be to use ranks from the different algorithms. The work discussed by Gao and Tan(116) proposes to use mixture modeling with the EM-Framework to convert scores into propababilities and thus deal with differences in scales in a safe way.

2.3.5.2 Model combination

One more issue in creating outlier detection ensembles, according to Aggarwal(14) is using a function to combine the models. Given that you have a set of r normalized scores $Score_i(\bar{X})$ for the point $\bar{X}$, it is necessary to use a function.

Aggarwal(14) lists the most common functions to combine models:

(i) Maximum function:

  (a) This is one of the most common functions used for combining ensemblar scores both in implicit (LOF and LOCI parameter tuning) and explicit ensemble models.

  (b) One variation of this model is to use the ranks instead of the scores in the combination process. This was proposed in feature bagging(105).

  (c) Different data points need to have the same number of components in the ensemble in order to be compared meaningfully.

(ii) Averaging Function:

  (a) The model scores are averaged over the different components of the ensemble.

  (b) If the individual components of the ensemble are poorly derived models, then the irrelevant scores from many different components will dilute the overall outlier score.

  (c) This approach has been used extensively, and it has the advantage of robustness because it reduces the variance of the overall prediction.

  (d) Variance reduction often results in superior performance.

  (e) Both the feature bagging(105) and the HiCS method(104) use this approach.

(iii) Damped averaging:

  (a) A damping function is applied to the outlier scores before averaging, in order to prevent it from being dominated by a few components.

  (b) A damping function could be the square root or the logarithm, for example.

  (c) The use of the product of the outlier scores (or geometric averaging) could be interpreted as the averaging of the logarithm of outlier scores.

(d) This approach is appropriate for outlier scores that are interpreted as probabilities or fit values, since the logarithms of the probabilities correspond to log-likelihood estimates.

(iv) Pruned averaging and aggregates:

(a) In this method, low scores are pruned and the outlier scores are either averaged or aggregated (summed up) over the relevant ensembles.

(b) The goal is to prune the irrelevant models for each data point before computing the combination score.

(c) The pruning can be performed by either using an absolute threshold on the outlier score or by picking the top-k models for each data point and averaging them.

(d) When using absolute thresholds, it is necessary to normalize the scores from the different ensemble components. The work proposed by Aggarwal(109) advocates the conversion of scores to standardized Z-values and then uses a threshold of 0 on the scores.

(e) Aggregating is more appropriate than averaging, since it implicitly counts the number of ensemble components in which a data point is relevant. The point is more relevant in a greater number of ensemble components when it has a greater tendency to be an outlier. The aggregated score, called thresh in the work of Aggarwal(109), combines the benefits of maximization and averaging.

(f) Aggarwal(14) guarantees this approach can often provide more robust results.

(v) Result from last component executed:

(a) Sometimes used in sequential ensembles(110), in which each component of the ensemble successively refines the data set and removes the obvious outliers.

(b) The normal model is constructed on a data set from which outliers are removed and the model is more robust.

(c) The goal of each component of the sequential ensemble is to successively refine the data set.

(d) The score from the last component is the most appropriate one to be used, in Aggarwal's(14) opinion.

As Aggarwal(14) points out, a combination function may be dependent on the structure of the ensemble, specially if the goal is to refine the whole dataset

or to understand behavior of a specific segment in this dataset. The author also notes that scores of outlier points tend to be far more unstable than those of inlier points. That's why, he says, maximization functions usually improve the overall performance. The author also suggests combining maximization function and averaging function to get more robust results.

2.4 Technique's summary and their relationship to this work

Below is a summarized table with techniques described above, indication if they were used in this work and why.

Table 2.2: Outlier detection technique summary table and their relationship to the ensemble creation.

Technique	**Used in the tests**	**Reason why**
Gaussian technique	No	Unidimensional. Makes it difficult to deal with multidimensional problems. Also is parametric and assumes normal distributions.
Histogram technique	No	Unidimensional. Makes it difficult to deal with multidimensional problems.
Boxplot technique	No	Unidimensional. Defines outliers based on the Interquartile range, which can be relative and label outliers in a wrong maner in some cases.
Regression technique	No	works on a weak correlation between two dimensions.
KNN	No	Weak detector based on distance between points only. Depends on numeric datasets and on number of K.
Local Outlier Factors	Yes	More robust detector than KNN because takes into account clustering relationship in groups and between groups.
COF	No	Similar to the LOF outlier detector.
ODIN	No	Low performance on time complexity
MDEF	No	Low performance for multidimensional datasets
DBSCAN	Yes	Density-based high performance algorithm. Works well on detecting clusters in different shapes.
ROCK	Yes	Detector specialized for categorical variables.
SNN	No	Similar to DBSCAN algorithm but low performance for dense datasets.
WaveCluster	No	Low performance on multidimensional datasets.
SOM	No	Non-straightforward interpreation of clusters and comlex implementation.
K-Means	Yes	Easy interpretation and high performance on dense datasets.
CBLOF	No	Similar to LOF but has low performance for dense datasets.

3
Techniques and methodologies

This chapter will present the approach used to obtain answers to problems described in Chapters 1 and 2. This work has been divided in three main phases, as described below.

3.1
Data preparation

As discussed in Chapter 2, the problem of Outlier Dectetion can be considered as an unsupervised one and thus lacks labels. The lack of labels makes it impossible to actually stablish a ground-truth to which we can compare the model created. That´s why although this ensemble outlier detection model was built for the purposes of Marketing Science datasets, it was tested with general datasets, i.e. datasets belonging to different contexts.

3.1.1
Understanding training data

Data understanding began with data collection. And data used for the experiments were provided by Outlier Detection DataSets (ODDS)(127) database, a data repository that´s been actively developed and growing since the summer of 2016, according to the owner itself: Shebuti Rayana(127). The website provides a large collection of outlier detection datasets with ground-truth. These datasets described in Table 3.1 were used for training and modeling.

Datasets made available by ODDS came in separate parts: inliers and outliers separated. So there was a need to clean and treat them. That's why only a limited number of datasets was used. In the future, new contexts will be added to newer versions of the ensemble model.

Variables present in each dataset varied among three types: ID variables, behavior variables and label variables (outlier classes). Classes between outliers and inliers were never balanced, which added a new level of difficulty to model creation. But since class unbalance is a common characteristic to the outlier detection problem, they were not evened.

Bases obtained from ODDS are UCI Machine Learning Repository's(128) bases, treated to be used in classification, clusterization and outlier detection

Table 3.1: For training and modeling the following datasets were used:

Name	Context	Rows	Dimensions	% Outliers
Glass	Contains attributes regarding several glass types (multi-class).	214	9	9 (4.2%)
Wine	Results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultures.	129	13	10 (7.7%)
BreastW	Measurements for breast cancer cases.	683	9	239 (35%)
Letters	Subsampled data from 3 letters to form the normal class and randomly concatenated pairs of their dimensionality doubles.	1600	32	100 (6.25%)

problems, so they were not treated any further. Main changes made were: treatment of missing values, transformation of categorical values into dummy variables and splitting of data into training, test and validation datasets.

The computer configuration used to train and create the model was a very simple one. One MacOS MacPro with an Intel Core i7 processor of six cores of 8th generation and 2,6 GHz; a Radeon Pro 560X video board with 4GB GDDR5 memory; and a 16 GB 2400 MHz of DDR4 RAM memory. Software used to build the model was R 3.5.1 binary for OS X 10.11 (El Capitan) built also on GNU Fortran 6.1 for OS X 10.11. UI software used was RStudio 1.1.456 - Mac OS X 10.6+ (64-bit).

3.1.2 Understanding testing data

After the ensemble model was created, it was applied in four Marketing Sciences datasets. These were described in Table 3.2.

For matters of privacy protection and company Non-disclosure Agreements, the names of the companies could not be revealed. But since the discussion regarding possible impact for those companies was considered more important, this was not a reason to keep this information outside this work.

Table 3.2: Marketing Science datasets used after model building:

Name	Context	Rows	Dimensions
Dataset1	Airline tickets purchase data from a Brazilian Travel company.	2106971	11
Dataset2	Card transactions data from a Multinational Issuer company.	35975	11
Dataset3	Test drive lead data for Car Brand A from a Multinational Car Dealer company.	1113546	11
Dataset4	Clothing purchase data from a Multinational Retailer and E-commerce company.	5764888	11

3.2 Algorithm selection phase

As described in Chapter 2, when building an ensemble, one has to carefully choose the algorithms it's going to be built upon. And since the objective of the approach was to build a sequential model-centric ensemble, algorithm selection was a very important step.

Fifteen different techniques were tested during the algorithm selection phase. Amongst which ten of those were the following:

(i) K-means;

(ii) Local Outlier Factors;

(iii) Isolation Forests Ensemble;

(iv) Global Local Outlier Score from Hierarchies Algorithms;

(v) Local Correlation Integral Ensemble;

(vi) Local Density-Based Outlier Detection;

(vii) ROCK Outlier Detection;

(viii) Angle-based outlier detection;

(ix) Subspace outlier detection;

(x) Feature bagging-based outlier detection.

Those models and detectors were chosen among so many others because of characteristics detailedly explained in Chapter 2 regarding its strengths and weaknesses. Most detectors involved subspace local search or were already ensembles themselves.

First step was to use each algorithm or ensemble to be tested individually. For this, the following functions were used: *Func.FBOD, Func.SOD, Func.ABOD, rockCluster, ldbod, LOCI, hdbscan, hclust, lof, iForest* and *kmeans*.

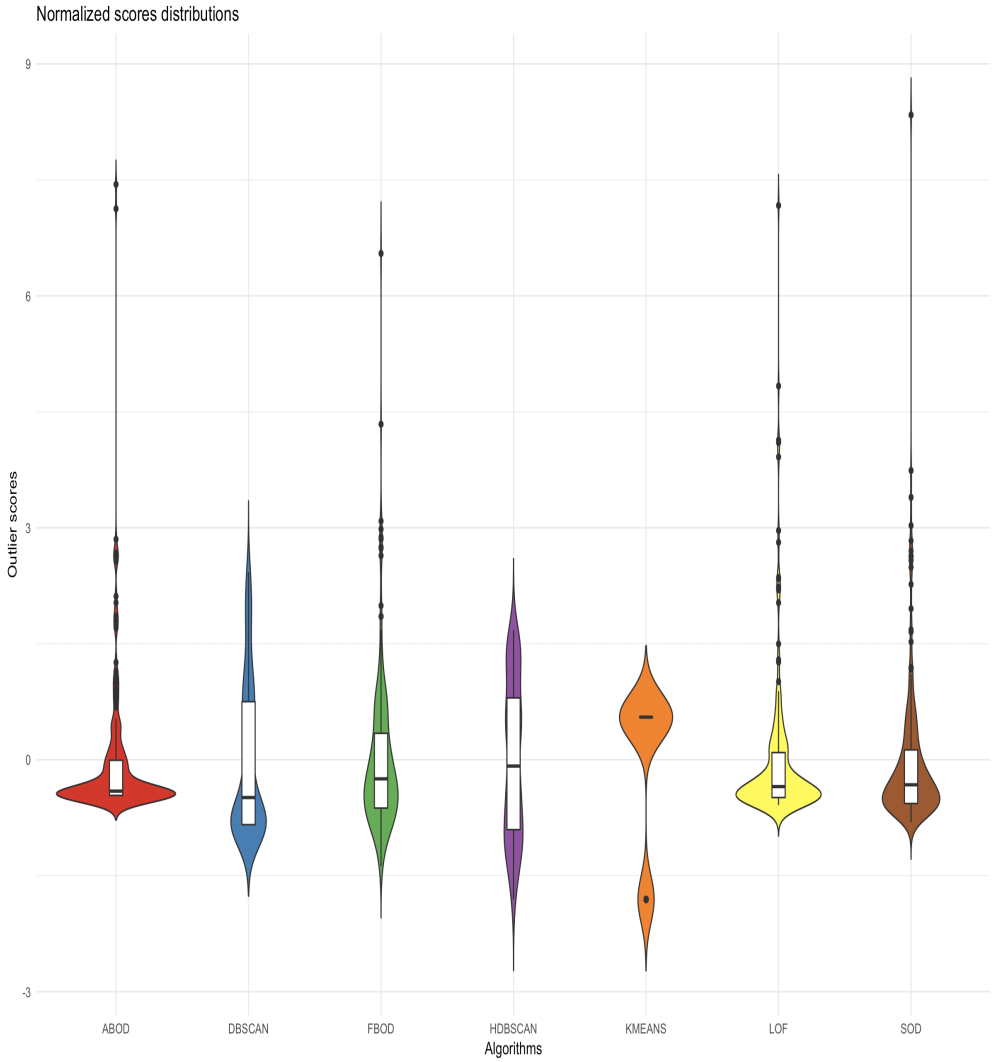

Figure 3.1: Normalized scores for seven algorithms tested. Dispersion of the boxplots makes it evident the very different behaviors of the detectors. The density curve makes it clear how each detector concentrated each classification.

Next step was selecting the best algorithm parameters among all parameters available. No feature selection was done, because most of the algorithms would do it for themselves using a number of different techniques. Process for parameter selection was done as following and based on the work of Ha and Nguyen(115):

1. Each algorithm was run with different K-fold samples of the same dataset a total of 100 times all with all variables available in the dataset;

2. For each run, accuracy and AUC curve were measured.

3. Algorithms were ordered by AUC score obtained;

4. Algorithms with performance below 0.75 were discarded.

To evaluate the process of algorithm selection, outlier scores generated by the individual detectors were separated in Train (60%), Test (20%) and Validation (20%) sets. A Stochastic Gradient Boosting model was run over all outlier scores in order to evaluate their accuracy. Parameters used for the evaluation model were:

1. Resampling: Cross-Validated (10 fold, repeated 10 times);

2. Random and different values used for number of iterations or the number of leaves;

3. Random and different values used for complexity of the tree or interaction depth;

4. Random and different values used for learning rate or shrinkage.

Prediction run by the GBM model was used to create a meta-learner - the ensemble model itself. The ideia was to understand how good predictors the detectors had been in learning the true outliers. And the reason why a classification algorithm was used in this phase was because there was a need to eliminate all possible bias when selecting the algorithms. Also scores from all detectors tested were normalized in order to account for their different ranges, as indicated in picture 3.2. Measures of evaluation used for the algorithms are described in Table 3.3.

After 1000 trials, only 4 detectors remained in the final model, as mentioned in Table 3.4.

Table 3.3: Performance evaluation metrics:

Name	Formula	Definition
Accuracy	$ACC = \frac{TP+TN}{OP+ON}$	Hits proportion, adding correct and wrong outlier cases. Returns values between 0 and 1, being 0 the worst case and 1 the best.
Sensitivity or True Positive Rate	$TPR = \frac{TP}{TP+FN}$	Capacity of the model to define as outliers the true outliers. Returns values between 0 and 1, being 0 the worst case and 1 the best.
Specificity or True Negative Rate	$TNR = \frac{TN}{FP+TN}$	Capacity of the model to define as inliers the true inliers. Returns values between 0 and 1, being 0 the worst case and 1 the best.
Precision or Positive Predictive Value	$PPV = \frac{TP}{TP+FP}$	Proportion of true predicted outliers among all predicted points. Returns values between 0 and 1, being 0 the worst case and 1 the best.
F1 Score	$F1 = 2 * \frac{PPV * TPR}{PPV+TPR}$	Harmonic mean between precision and sensitivity, in other words; a single measure in which you evaluate if total positively predicted cases are really positive and if they correspond to a significant part of total true positive points. Returns values between 0 and 1, being 0 the worst case and 1 the best.

Table 3.4: Detectors evaluated for the final model

Name	AVG. RMSE	Accuracy
LOF	17.8	0.7563333
FBOD	15.95	0.825
SOD	14.9	0.725
HDBSCAN	13.2	0.827

3.3 Model building phase

Model building phase involved the search of a model capable of better separating outliers and inliers in Multi-class datasets. To maintain comparability,

the same data was used both to train, create the model and evaluate its results. The objective of this approach was to build an ensemble capable of correctly separating outliers from inliers in different contexts so that when used in the Marketing Science context, it would hopefully have a performance above average for most purchase/transaction datasets.

The following steps were used as procedure to the ensemble creation:

1. Algorithm selection 3.2 and evaluation;
2. Discarding of low performance algorithms;
3. Creation of ensemble order for the algorithms;
4. Final evaluation.

The algorithm that describes this work is Algorithm 4.

Algorithm 4 Sequential model-centric ensemble

```
1:  procedure ENSEMBLE PROCESS(Dataset)
2:      j ← 1                                                    ▷ O (1)
3:      for j ← 1...n do                                         ▷ O (n)
4:          Alg_n1 ← newAlgorithmBasedOnPastExecutions           ▷ O (1)
5:          f_n ← newDatasetFromDbasedOnPastExecutions           ▷ O (1)
6:          f_j ← Alg_n1                                         ▷ O (1)
7:          Alg_n2 ← newAlgorithmBasedOnPastExecutions           ▷ O (1)
8:          f_n ← newDatasetFromDbasedOnPastExecutions           ▷ O (1)
9:          f_j ← Alg_n2                                         ▷ O (1)
10:         outliers ← scoreFromAlg1                             ▷ O (1)
11:         outliers ← scoreFromAlg2                             ▷ O (1)
12:         if scoreFromAlg1 = scoreFromAlg2 then                ▷ O (1)
13:             outliers ← newScoreFromBothAlg                   ▷ O (1)
14:         j ← j + 1                                            ▷ O (1)
15:     outliers ← combinationOfPreviousExecutions               ▷ O (1)
```

3.3.1 Detector ordering phase

Since the ensemble was meant to be a model-centric sequential one, algorithms were ordered according to their performances and each run depended on results from the previous run. According to Aggarwal(14), in a model-centric sequential outlier detection ensemble, algorithms should be prefferably ordered so the less accurate goes first and more robust detectors come afterwards and have the chance to improve the work done before.

In order to follow that, detectors chosen in the previous phase were ordered in an order inverse to their performance. It means that the less accurate was put

first, then the next one and so on. Aggarwal (14) explains that more powerful detectors should be the last to run on a sequential model-centric ensemble in order to account for previous mistakes made by more simple detectors.

Before following this approach though, detectors chosen were used in a random order on 100 trials in order to predict the best order for the ensemble. Results were worse than those obtained by following Aggarwal's suggestion on ordering them by lowest performance. So methodology followed below can be considered a development of Aggarwal's approach.

3.3.2 Model combination phase

For the meta-model created, scores generated by the first detector were compared to the scores generated by the second detector and then analyzed. If data points had been defined by both detectors as outliers, then a high probability would be given for the score. If only one detector had labeled the point an outlier, then a smaller probability was assigned to the score and so on.

After all detectors were run, all scores would have been compared to each other, each in its own order. Outlier labels were lastly assigned according to high probabilities of points having been assigned as "outliers" by all detectors.

3.4 Method Creation Outline

To summarize the methodology described previously, below follows a visual outline of the ensemble creation method. It was comprised in four stages: the first explored the data available to remove identification dimensions and keep outlier labels; the second explored the usage of different base detectors on the training datasets as well as classified them by performance; the third explored different orders for the detectors and the last phase explored different normalization methods and combinations for the ensemble.

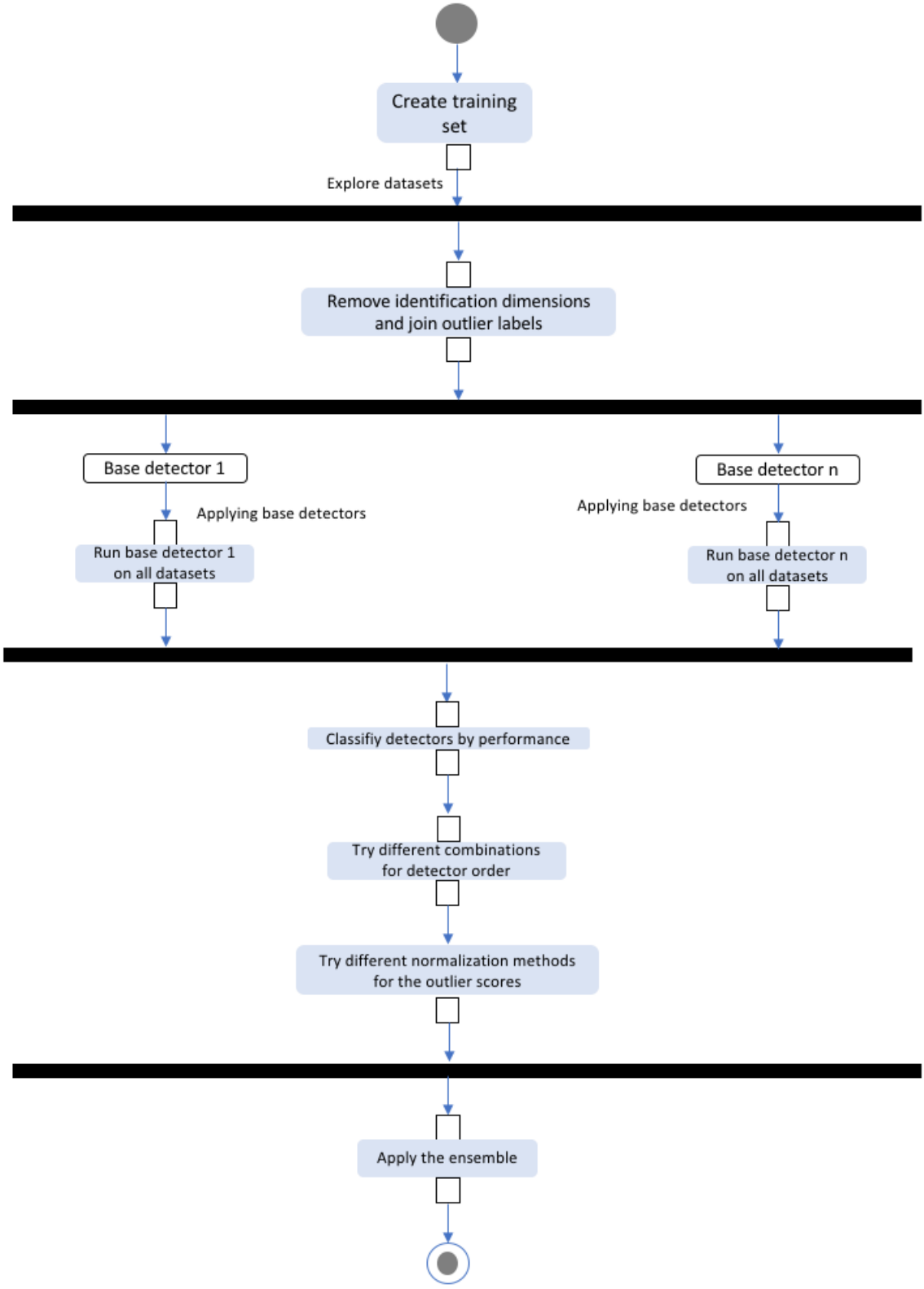

Figure 3.2: Method development outline.

4
Experiments and results

In this chapter, results obtained after the ensemble model - described in Chapter 3 - was created will be detailed. The objective when evaluating results was getting to an ensemble that was more accurate than detectors individually and that had a bigger area under the curve.

To evaluate the performance of the method, a comparison was made between several base outlier detectors, and also clustering and classification algorithms that are often used as outlier classifiers, like K-means, Local outlier Factors, angle-base outlier detectors etc. A second comparison was made between the model and other outlier detecion ensembles like iForest and LOCI. The comparison was meant to assess the quality of the classification AUC and Accuracy. Keeping in mind though that this model will be later used to detect outliers in an unsupervised way without the use of any ground-truth. So there's an assumption here that whatever results were evaluated for the classification model will prove themselves correct in an unsupervised problem.

4.1
Performance of the Model

In order to compare the models and the detectors, several measures were adopted such as Accuracy, Area Under the Curve (AUC), F1 Score, Predicted Positive Condition Rate, Specificity and Sensitivity. All base detectors and ensembles tested are also available in the R environment[1].

First technique used in the ensemble was HDBSCAN, which is a clustering algorithm that extends the original DBSCAN by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based in the stability of clusters. For datasets Wine and Glass performance both in AUC and Accuracy between HDBSCAN and the ensemble were quite similar. For datasets BreastW and Letters though, performance of the ensemble was superior to the original detector.

Second technique used in the ensemble was the Local Outlier Factor with a K of 10. Multiple values of K had been tested previously in the detectors selection

[1] R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. https://www.r-project.org/

Table 4.1: Datasets used as base for building the ensemble and performance of the methods:

Name	**Method**	**Accur.**	**AUC**	**F1 Score**	**Pred. Pos. Cond. Rate**	**Spec.**	**Sens.**
Glass	SOD	0.9	0.72	0.2	0.05	0.95	0.22
	FBOD	0.89	0.57	0.2	0.05	0.95	0.22
	LOF 10	0.8	0.78	0.07	0.16	0.93	0.22
	HDBSCAN	0.81	0.8	**0.25**	**0.3**	0.82	**0.28**
	ENSEMBLE	**0.92**	**0.83**	0.05	0.11	**0.97**	0.26
Wine	SOD	0.89	0.68	0.05	0	0.94	0
	FBOD	**0.9**	0.63	0.05	0.1	0.95	0.1
	LOF 10	**0.9**	0.53	0.03	**0.23**	**0.97**	0.22
	HDBSCAN	0.89	0.51	0.06	0.18	0.94	0.2
	ENSEMBLE	0.88	**0.89**	**0.1**	0.09	**0.97**	**0.24**
BreastW	SOD	**0.89**	0.72	0.07	0.05	0.06	**0.91**
	FBOD	0.87	0.57	0.2	0.09	0.1	0.35
	LOF 10	0.19	0.52	0.35	0	0.35	0
	HDBSCAN	0.37	0.47	0.53	**0.92**	0.003	0.82
	ENSEMBLE	**0.89**	**0.79**	**0.6**	0.4	**0.9**	0.35
Letters	SOD	0.77	0.55	0.15	0.03	0.3	0.29
	FBOD	0.75	0.57	0.24	0.01	0.46	0.2
	LOF 10	0.05	0.35	0.35	0	**0.78**	0
	HDBSCAN	0.55	0.52	0.6	**0.9**	0.05	**0.68**
	ENSEMBLE	**0.79**	**0.68**	**0.74**	0.1	0.65	0.59

phase and K of 10 revealed to be the best choice for the datasets trained. Local Outlier Factor was described in Section 2.2.2.2. In Table 4.1 it's possible to see that LOF had a performance quite inferior to the final ensemble for datasets BreastW and Letters, while for dataset Glass it was similar and for dataset Wine it was nearly superior to the final ensemble.

Third technique used was the Feature Bagging-based Outlier Detection(105), which is an ensemble per se that combines results from multiple outlier detection algorithms that are applied using different set of features. The FBOD detector had a similar performance to the final ensemble in all datasets, with a slight superior performance in dataset Wine.

Fourth and last technique used in the ensemble was the Subspace Outlier Detection(129). This detector uses a robust subspace method for detecting such inner outliers in a given dataset, which uses two dimensional-projections: detecting outliers in subspaces with local density ratio in the first projected dimen-

sions; finding outliers by comparing neighbour's positions in the second projected dimensions. Each point's weight is calculated by summing up all related values got in the two steps projected dimensions, and then the points scoring the largest weight values are taken as outliers. This detector had the best performance among all detectors and was outperformed by the ensemble only for the Letters dataset, reason why it was included in the final model.

It's possible to see in Image 4.1 that performance though similar for some detectors and the final ensemble - for the dataset Glass - meant different labels for data points. This comparison won't be possible when detecting outliers in real-world datasets though, because a ground-truth won't be available. That's why all bias had to be removed from the model construction in order to account for outlier labeling in future datasets.

Figure 4.1: Data points colored by outliers and inliers as per the real dataset, tested detectors and the final ensemble model.

Also, regarding the Bias-Variance trade-off theory discussed previously in Chapter 2, both the ensemble and the algorithms tested had a low variance but a high bias. What this means is that they were too well adjusted for the datasets they were trained for. So even when performing multiple times the same

algorithm in the same dataset, results would not vary much. But when applying the whole ensemble from one dataset to the other, results showed that bias and overfitting were indeed present. This is something to be worked on in future works.

A comparison between AUC curves for detectors used and the final ensemble was also provided in Figure 4.2.

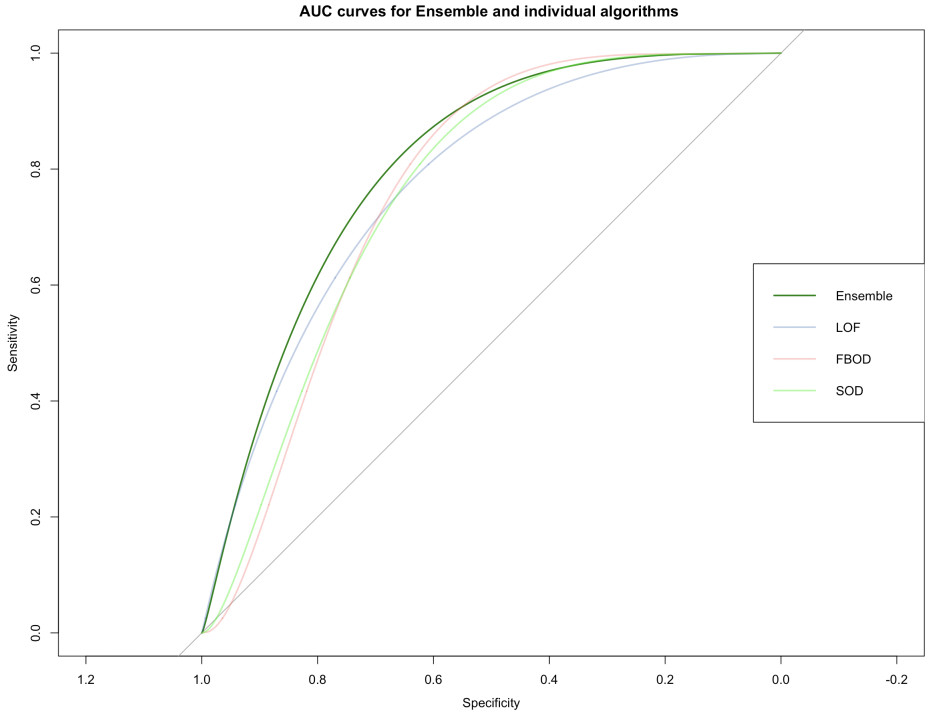

Figure 4.2: AUC curves colored by algorithms.

4.2 Expected Lift

This work compared pre-detection of outliers and post-detection of outliers with the following metrics:

(i) Incremental Conversions:

(a) Conversions may be any event performed by the user when being tracked in the study. A landing page view, an add to cart or a purchase. In this case, the event considered were purchases.

(b) In this case, only incremental conversions were considered. In other words, only conversions that happened because of the ad. Meaning also: difference in conversions between Control and Exposed groups.

(ii) Incremental Sales:

(a) Sales is the value of each conversion or purchase. So, whatever currency was tracked by the study, Sales would be the total amount in value of all purchases.

(b) In this case, only incremental sales were considered. In other words, only the value of incremental conversions. Meaning also: difference in sales between Control and Exposed groups.

(iii) Incremental ROAS:

(a) ROAS is the Return on Ad Spend. Similar to ROI (Return on Investment), but only considering spend in media.

(b) In this case, only incremental ROAS was considered. In other words, only return that happened because of the incremental conversions.

(c) Incremental ROAS is calculated dividing incremental Sales by total investment.

Those metrics were chosen because they are the metrics used in Lift Studies run by advertisers on Facebook.

4.3 Testing in real-world datasets

So after ensemble creation and testing, the model was also tested in real Marketing Science datasets. Details were provided in Table 4.2. In the case of the Retail Dataset, for example, 7% of outliers were removed after running the Ensemble Model. And after excluding from the dataset what the model considered as outliers, difference in Lift increased from 173 conversions to 310 conversions. What this means is that prior to outlier detection, difference between conversions in the Control Group and conversions in the Exposed Group was 173. But after removing outlier users from the dataset (post-detection), difference between Control Group and Exposed Group was 310 conversions. Difference in Sales was even bigger. Pre-detection (considering outliers), difference in Sales between groups was $21.800, but post-detection it was $107.419, around 5 times bigger. ROAS, being a relative of Sales, also increased from 0.89x to 4.38x.

The Test Drive dataset had a pretty similar result to the Retail dataset. And removing 6% of outliers meant doubling Sales and also ROAS. In this case, however, for an Auto company, saying that your ads are driving double the amount of sales you thought they were has a bigger impact, since the price of a car is more expensive.

The Card dataset only had 1% of its users removed and a small increase in Lift. This also meant a little difference in Sales and ROAS. One thing worth

Table 4.2: Final results for Marketing Science datasets after running the ensemble model

Dataset	**% outliers removed**	**Lift**	**Pre-detection**	**Post-detection**
Retail dataset	7%	Conversions	173	310
		Sales	$21.800	$107.419
		ROAS	0.89x	4.38x
Travel dataset	2%	Conversions	1201	4515
		Sales	$10.212	$85.451
		ROAS	0.1x	0.23x
Card dataset	1%	Conversions	615	656
		Sales	$525	$612
		ROAS	1.01x	1.17x
Test drive dataset	6%	Conversions	102	223
		Sales	$6000000	$12400450
		ROAS	1.2x	2.48x

noticing though is that the Card Issuer company that provided this dataset didn't know for sure how much "revenue" a converted user meant nor could they reveal any actual card transaction data, so this work ended up estimating along with the company how much "worth" a user is to a card company on average.

The Travel dataset also had a small amount of users removed by the Ensemble model, but in this case it meant a big difference in Sales and in ROAS. And this holds the hypothesis previously stated that outliers were indeed impacting Lift results. The Travel company that provided the dataset had already confirmed they deal with outliers on a daily basis, since most travel agencies buy air travel tickets using the name of a person of the company and not the name of the company itself. So this "outlier" user which is in fact just a travel bureau ends up getting mixed among the regular users. If this outlier ends up belonging to the Control group, it can end up impacting the Lift calculation and the difference in Sales.

So, generally speaking, all datasets were positively impacted by outlier removal and all showed increase in Lift results between pre-detection and post-detection.

5
Conclusion and future work

5.1
Outcomes obtained

This work presented a method based on outlier detection ensemble, which uses scores generated by base detectors and applies a probability-based comparison combining their scores. Finally, it gets to a consensus of points that should be labed as outliers in the process. The idea was to improve efficacy of base detectors already used in Marketing Science datasets related to performance advertisers. Search for improvement was done using outlier detection techniques and ensembles to get to a model that would show better results.

Research and results obtained show that clustering and outlier detection techniques alone are not sufficient to account for true outliers in most datasets. Final model created from some of the most reliable existing base detectors and ensembles showed good results with an average 3.5% increase in the AUC and average 7% increase in accuracy.

As for results in Marketing Science datasets, after running the ensemble, it was possible to see that increase in Lift of incremental conversions, sales and ROAS was bigger the more outliers the ensemble had to clean away. This reinforces the hipothesis this work had made upon the existance of outliers hiding potential lift in conversions for datasets with a high outlier presence. Although when discussing the bias-variance trade-off, this work achieve better results in lowering variance than in lowering bias. As results in the previous chapter point, for the Wine dataset it was not possible to achieve good results neither with the detectors nor with the ensemble, which shows the model was still very oriented to the datasets it trained with.

5.2
Future works

Future works should focus on lowering the bias as well as the variance, or in other words, being context-free and able to deal with datasets from most industries. To do this, one direction could be to focus on a more robust way to

keep accuracy and AUC results while preventing from having a high bias. Maybe testing different approachs on the model combination.

Another direction on building future works could be working with base detectors that do not treat outliers on a binary sense but more on a probability of data points being an outlier. Because even though scores are continuous numbers that classify outliers from 0 to 1, when labelling outliers and measuring performance, most works treat them in a binary way: either the point is or isn't an outlier. Future works could focus on the probability a point is an outlier and how this could affect the overall results.

Bibliography

[1] EMARKETER. **Worldwide Retail and Ecommerce Sales: eMarketer's Updated Forecast and New Mcommerce Estimates for 2016-2021**, 2018. Acesso em: Maio de 2018.

[2] EMARKETER. **Latin America Ad Spending Summary 2018**, 2018. Acesso em: Abril de 2018.

[3] SMALLWOOD, B.. **Resisting the siren call of popular digital media measures: Facebook research shows no link between trendy online measures and ad effectiveness**. Journal of Advertising Research, 56:7, 2016.

[4] NIELSEN. **Shopper Path to Purchase: The Three Biggest Decisions You Can Influence**, 2017. Acesso em: Maio de 2018.

[5] BELL, D. R.; CORSTEN D.; KNOX G.. **From point of purchase to path of purchase: Hw preshopping factors drive unplanned buying**. Journal of Marketing, 75:31–45, 2011.

[6] IPSOS. **Growing sales by understanding and influencing shoppers along their path to purchase**, 2017. Acesso em: Maio de 2018.

[7] OPPENHEIMER, M.. **Internet cookies: When is permission consent?** University of Baltimore Law, 85, 2006.

[8] UNIVERSITY, G. S.. **Internet Cookies as a Surveillance Technique**, 2010. Acesso em: Maio de 2018.

[9] ZHAO, K.; MAHBOOBI S. H.; BAGHERI S. R.. **Revenue-based attribution modeling for online advertising**. Computer Statistical Data Analysis, 80:223–239, 2017.

[10] UK, I.. **Attribution Whitepaper**, 2015. Acesso em: Maio de 2018.

[11] AGGARWAL, C.C.. **Outlier Analysis**. Springer, New York, 2nd edition, 2013.

[12] TAN, P.; STEINBACH, M.; KUMAR, V.. **Introduction to Data Mining**. Pearson Education, New York, 1st edition, 2006.

[13] BIRKBECK UNI.. **A probabilistic multi-touch attribution model for online advertising**. CIKM, 2016.

[14] AGGARWAL, C. C.; SATHE, S.. **Outlier Ensembles: an Introduction**. Springer, New York, 1st edition, 2017.

[15] HAWKINS, D. M.. **Identification of Outliers**. Springer, New York, 1st edition, 1980.

[16] BEN-GAL, I.. **Outlier detection**. In: MAIMON & ROKACH, editor, DATA MINING AND KNOWLEDGE DISCOVERY HANDBOOK: A COMPLETE GUIDE FOR PRACTITIONERS AND RESEARCHERS, volumen 1, 2nd ed, chapter 7, p. 117–133. Springer, New York (NY, USA), 2010.

[17] CHANDOLA, V.; BANERJEE, A.; KUMAR, V.. **Anomaly detection: A survey**. ACM, 41:58, 2009.

[18] ESKIN, E.. **Anomaly detection over noisy data using learned probability distributions**. In: PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 255–262. Morgan Kaufmann Publishers Inc., 2000.

[19] DESFORGES, M.; JACOB, P.; AND COOPER, J. **Applications of probability density estimation to the detection of abnormal conditions in engineering**. In: PROCEEDINGS OF THE INSTITUTE OF THE MECHANICAL ENGINEERS, volumen 212, p. 687–703. 1998.

[20] WIKIPEDIA. **Normal Distribution**. Acesso em: Maio de 2018.

[21] SHEWHART, W. A.. **Economic control of quality of manufactured product**. D. Van Nostrand Company, 1931.

[22] BARNETT, V. AND LEWIS, T.. **Outliers in Statistical Data**. John Wiley, 1st edition, 1994.

[23] BARNETT, V.. **The ordering of multivariate data (with discussion)**. J. Royal Statis. Soc. Series A, 139:318–354, 1976.

[24] BECKMAN, R. J. AND COOK, R. D.. **Outlier analysis**. Technometrics 25, 2:119–149, 1983.

[25] LAURIKKALA, J., JUHOLA, M., AND KENTALA., E.. **Informal identification of outliers in medical data**. In: PROCEEDINGS OF THE 5TH INTERNATIONAL WORKSHOP ON INTELLIGENT DATA ANALYSIS IN MEDICINE AND PHARMACOLOGY, p. 20–24. 2000.

[26] HORN, P. S., FENG, L., LI, Y., AND PESCE, A. J.. **Effect of outliers and nonhealthy individuals on reference interval estimation**. Clinical Chem. 47, 12:2137–2145, 2001.

[27] SOLBERG, H. E. AND LAHTI, A.. **Detection of outliers in reference distributions: Performance of horn's algorithm**. Clinical Chem. 51, 12:2326–2332, 1983.

[28] WIKIPEDIA. **Linear Regression**. Acesso em: Maio de 2018.

[29] ABRAHAM, B. AND CHUANG, A.. **Outlier detection and time series modeling**. Technometrics 31, 2:241–248, 1989.

[30] ABRAHAM, B. AND BOX, G. E. P.. **Bayesian analysis of some outlier problems in time series**. Biometrika 66, 2:229–236, 1979.

[31] FOX, A. J.. **Outliers in time series**. J. Royal Statis. Soc. Series B 34, 3:350–363, 1972.

[32] AGARWAL, D.. **Detecting anomalies in cross-classified streams: A bayesian approach. knowl**. Inform. Syst. 11, 1:29–44, 2006.

[33] ZHIHUA Z.. **Ensemble Methods: Foundations and Algorithms**. Chapman and Hall/CRC, 1st edition, 2012.

[36] BICKEL, S.;SCHEFFER, T.. **Multi-view clustering**, 2004. Acesso em: Maio de 2018.

[37] MULLER, E.;GUNNEMANN, S. I. T.. **Discovering multiple clustering solutions: Grouping objects in different views of the data**, 2010. Acesso em: Maio de 2018.

[38] MULLER, E.;GUNNEMANN, S. T. I.. **Tutorial: Discovering Multiple Clustering Solutions Grouping Objects in Different Views of the Data**, 2012. Acesso em: Maio de 2018.

[39] STREHL,A.;GHOSH, J.. **Cluster ensembles: A knowledge reuse framework for combining multiple partitions**. Journal of Machine Learning Research, 3:583–617, 2001.

[40] AGGARWAL, C.C.. **A human-computer interactive method for projected clustering**. IEEE Transactions on Knowledge and Data Engineering, 16:448–460, 2004.

[41] HINNEBURG, A.;KEIM, D.;WAWRYNIUK, M.. **Hd-eye:visual mining of high-dimensional data**. IEEE Computer Graphics and Applications, 19:22–31, 1999.

[42] MOOSMANN, F.;TRIGGS, B.;JURIE, F.. **Fast discriminative visual codebooks using randomized clustering forests**. Neural Information Processing Systems, p. 985–992, 2006.

[43] LIU, F. T.;TING, K. M.;ZHOU, Z. H.. **Isolation Forest**. 2008. Acesso em: Maio de 2018.

[44] BRIEMAN, L.. **Bagging Predictors**. Machine Learning, 24:123–140, 1996.

[45] FREUNDAND, Y.; SCHAPIRE, R.. **A decision-theoretic generalization of online learning and application to boosting**. Computational Learning Theory, 1995.

[46] CLARKE, B.. **Bayes model averaging and stacking when model approximation error cannot be ignored**. Journal of Machine Learning Research, p. 683–712, 2003.

[47] CLARKE, B.. **Bayes model averaging and stacking when model approximation error cannot be ignored**. Machine Learning Journal,, p. 59–83, 1999.

[48] WOLPERT, D.. **Stacked generalization**. Neural Networks, 5:241–259, 1992.

[49] BRIEMAN, L.. **Randomforests**. JournalMachineLearningarchive, 45:5–32, 2001.

[50] HO, T.K.. **The random subspace method for constructing decision forests**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:832–844, 1998.

[51] DOMINGOS, P.. **Bayesian averaging of classifiers and the overfitting problem**. Relatório interno, ICML Conference, 2000.

[52] ZENKO, B.. **Is combining classifiers better than selecting the best one**. Machine Learning, p. 255–273, 2004.

[53] CHAWLA, N.; LAZAREVIC, A.;HALL, L.; BOWYER, K.. **Smoteboost: Improving prediction of the minority class in boosting**. PKDD, p. 107–119, 2003.

[54] JOSHI, M.;KUMAR, V.; AGARWAL, R.. **Evaluating boosting algorithms to classify rare classes: Comparison and improvements**. Relatório Interno 257-264, ICDM Conference, 2001.

[55] MICENKOVA, B.; MCWILLIAMS, B.; ASSENT, I.. **Learning outlier ensembles: The best of both worlds supervised and unsupervised**. Relatório interno, ACM SIGKDD Workshop on Outlier Detection and Description, 2014.

[56] MICENKOVA, B.; MCWILLIAMS, B.; ASSENT, I.. **Learning representations for outlier detection on a budget**. Relatório interno, 2014.

[57] ANDERSON, D.; FRIVOLD, T.; TAMARU, A.; AND VALDES, A.. **Next-generation intrusion detection expert system (nides), software users manual, beta-update release**. Sri–csl–95–07, SRI International, 1994.

[58] JAVITZ, H. S.; VALDES, A.. **The sri ides statistical anomaly detector**. In: PROCEEDINGS OF THE IEEE SYMPOSIUM ON RESEARCH IN SECURITY AND PRIVACY. IEEE Computer Society, 1991.

[59] HELMAN, P.; BHANGOO, J.. **A statistically-based system for prioritizing information exploration under uncertainty**. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, volumen 27, p. 449–466. Man, and Cybernetics, 1997.

[60] ENDLER, D.. **Intrusion detection: Applying machine learning to solaris audit data**. In: PROCEEDINGS OF THE 14TH ANNUAL COMPUTER SECURITY APPLICATIONS CONFERENCE, p. 268. IEEE Computer Society, 1998.

[61] HO, L. L.; MACEY, C. J.; HILLER, R.. **A distributed and reliable platform for adaptive anomaly detection in ip networks**. In: PROCEEDINGS OF THE 10TH IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS AND MANAGEMENT, p. 33–46. Springer-Verlag, 1999.

[62] YAMANISHI, K.; ICHI TAKEUCHI, J.. **Discovering outlier filtering rules from unlabeled data: Combining a supervised learner with an unsupervised learner**. In: PROCEEDINGS OF THE 7TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 389–394. ACM Press, 2001.

[63] YAMANISHI, K.; TAKEUCHI, J.-I.; WILLIAMS, G.; AND MILNE, P.. **Online unsupervised outlier detection using finite mixtures with discounting learning algorithms**. Data Min. Knowl, 8:275–300, 2004.

[64] FAWCETT, T.; PROVOST, F.. **Activity monitoring: noticing interesting changes in behavior**. In: PROCEEDINGS OF THE 5TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 53–62. ACM Press, 1999.

[65] MANSON, G.. **Identifying damage sensitive, environment insensitive features for damage detection**. In: PROCEEDINGS OF IES CONFERENCE. 2002.

[66] MANSON, G.; PIERCE, G.; WORDEN, K.. **On the long-term stability of normal conditions for damage detection in a composite panel**. In: PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON DAMAGE ASSESSMENT OF STRUCTURES. Cardiff, UK, 2001.

[67] MANSON, G.; PIERCE, S. G.; WORDEN, K.; MONNIER, T.; GUY, P.; ATHERTON, K.. **Long-term stability of normal condition data for novelty detection**. In: PROCEEDINGS OF THE CONFERENCE ON SMART STRUCTURES AND INTEGRATED SYSTEMS, p. 323–334. 2000.

[68] KRUEGEL, C.; VIGNA, G.. **Anomaly detection of web-based attacks**. In: PROCEEDINGS OF THE 10TH ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, p. 251–261. ACM Press, 2003.

[69] KRUEGEL, C.; TOTH, T.; KIRDA, E.. **Service specific anomaly detection for network intrusion detection**. In: PROCEEDINGS OF THE ACM SYMPOSIUM ON APPLIED COMPUTING, p. 201–208. ACM Press, 2002.

[70] ALLAN, J.; CARBONELL, J.; DODDINGTON, G.; YAMRON, J.; YANG, Y.. **Topic detection and tracking pilot study**. In: PROCEEDINGS OF THE DARPA BROADCAST NEWS TRANSCRIPTION AND UNDERSTANDING WORKSHOPS, p. 194–218. 1998.

[71] PARZEN, E.. **On the estimation of a probability density function and model**. Annals Math. Stat., 33, 1962.

[72] DESFORGES, M.; JACOB, P.; COOPER, J.. **Applications of probability density estimation to the detection of abnormal conditions in engineering.** In: PROCEEDINGS OF THE INSTITUTE OF THE MECHANICAL ENGINEERS, volumen 212, p. 687–703. 1998.

[73] SAYAD, S.. **Numerical variables**. Acesso em: Abril de 2018.

[74] ANDERSEN, ERLING B.. **Discrete Statistical Models with Social Science Applications**. North Holland, 1st edition, 1980.

[75] ESKIN, E.; LEE, W.; STOLFO, S.. **Modeling system call for intrusion detection using dynamic window sizes.** In: PROCEEDINGS OF DARPA INFORMATION SURVIVABILITY CONFERENCE AND EXPOSITION (DISCEX). 2001.

[76] ANGIULLI, F.; PIZZUTI, C.. **Fast outlier detection in high dimensional spaces.** In: PROCEEDINGS OF THE 6TH EUROPEAN CONFERENCE ON PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY, p. 15–26. Springer-Verlag, 2002.

[77] ZHANG, J.; WANG, H.. **Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance**. Knowl. Inform. Syst., 46:333–355, 2006.

[78] KNORR, E. M.; NG, R. T.. **A unified approach for mining outliers.** In: PROCEEDINGS OF THE CONFERENCE OF THE CENTRE FOR ADVANCED STUDIES ON COLLABORATIVE RESEARCH., volumen 11. IBM Press, 1997.

[79] KNORR, E. M.; NG, R. T.. **Algorithms for mining distance-based outliers in large datasets.** In: PROCEEDINGS OF THE 24RD INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES., p. 392–403. Morgan Kaufmann Publishers Inc., 1998.

[80] KNORR, E. M.; NG, R. T.. **Finding intensional knowledge of distance-based outliers**. VLDB J., p. 211–222, 1999.

[81] KNORR, E. M.; NG, R. T.; TUCAKOV, V.. **Distance-based outliers: Algorithms and applications**. VLDB J., 8:237–253, 2000.

[82] WEI, L.; QIAN, W.; ZHOU, A.; JIN, W.. **Hot: Hypergraph-based outlier test for categorical data.** In: PROCEEDINGS OF THE 7TH PACIFIC-ASIA CONFERENCE ON KNOWLEDGE AND DATA DISCOVERY., p. 399–410. 2003.

[83] BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J.. **Optics-of: Identifying local outliers**. In: PROCEEDINGS OF THE 3RD EUROPEAN CONFERENCE ON PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY, p. 262–270. Springer-Verlag, 1999.

[84] BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J.. **Lof: Identifying density-based local outliers**. In: PROCEEDINGS OF THE ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, p. 93–104. ACM Press, 2000.

[85] TANG, J.; CHEN, Z.; CHEE FU; A. W.; W.CHEUNG, D.. **Enhancing effectiveness of outlier detections for low density patterns**. In: PROCEEDINGS OF THE PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 535–548. 2002.

[86] HAUTAMAKI, V.; KARKKAINEN, I.; FRANTI, P.. **Outlier detection using k-nearest neighbour graph**. In: 3, editor, PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, p. 430–433. IEEE Computer Society, 2004.

[87] PAPADIMITRIOU, S.; KITAGAWA, H.; GIBBONS, P. B.; FALOUTSOS, C.. **Loci: Fast outlier detection using the local correlation integral**. Relatório Interno IRP-TR-02-09, Intel Research Laboratory, 2002.

[88] ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X.. **A density-based algorithm for discovering clusters in large spatial databases with noise**. In: Simoudis, E.; Han, J. F. U., editor, PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 226–231. Eds. AAAI Press, 1996.

[89] GUHA, S.; RASTOGI, R.; SHIM, K.. **Rock: A robust clustering algorithm for categorical attributes**. Inform. Syst. 25, 5:345–366, 2000.

[90] ERTOZ, L.; STEINBACH, M.; KUMAR, V.. **Finding topics in collections of documents: A shared nearest neighbor approach**. In: CLUSTERING AND INFORMATION RETRIEVAL, p. 83–104. 2003.

[91] YU, D.; SHEIKHOLESLAMI, G.; ZHANG, A.. **Findout: Finding outliers in very large datasets**. Knowl. Inform. Syst. 4, 4:387–412, 2002.

[92] SHEIKHOLESLAMI, G.; CHATTERJEE, S.; ZHANG, A.. **Wavecluster: A multi-resolution clustering approach for very large spatial databases**. In: PROCEEDINGS OF THE 24RD INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, p. 428–439. Morgan Kaufmann Publishers Inc., 1998.

[93] SMITH, R.; BIVENS, A.; EMBRECHTS, M.; PALAGIRI, C.; SZYMANSKI, B.. **Clustering approaches for anomaly-based intrusion detection**.

In: PROCEEDINGS OF THE INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS, p. 579–584. ASME Press, 2002.

[94] PIRES, A.; SANTOS-PEREIRA, C.. **Using clustering and robust estimators to detect outliers in multivariate data**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ROBUST STATISTICS. 2005.

[95] OTEY, M.; PARTHASARATHY, S.; GHOTING, A.; LI, G.; NARRAVULA, S.; PANDA, D.. **Towards nic-based intrusion detection**. In: PROCEEDINGS OF THE 9TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 723–728. ACM Press, 2003.

[96] ESKIN, E.; ARNOLD, A.; PRERAU, M.; PORTNOY, L.; STOLFO, S.. **A geometric framework for unsupervised anomaly detection**. In: PROCEEDINGS OF THE CONFERENCE ON APPLICATIONS OF DATA MINING IN COMPUTER SECURITY, p. 78–100. Kluwer Academics, 2002.

[97] MAHONEY, M. V.; CHAN, P. K.. **Learning rules for anomaly detection of hostile network traffic**. In: PROCEEDINGS OF THE 3RD IEEE INTERNATIONAL CONFERENCE ON DATA MINING, p. 601. IEEE Computer Society, 2003.

[98] JIANG, M. F.; TSENG, S. S.; SU, C. M.. **Two-phase clustering process for outliers detection**. Patt.Recog. Lett. 22, 6-7:691–700, 2001.

[99] HE, Z.; XU, X., ;DENG, S.. **Discovering cluster-based local outliers**. Pattern Recog. Lett. 24, 9-10:1641–1650, 2003.

[100] BENTLEY, J. L.. **Multidimensional binary search trees used for associative searching**. Communications of the ACM. 18, 9:509, 1975.

[101] RUSSELL, S. J.; NORVIG, P.. **Artificial Intelligence: A Modern Approach**. 3rd edition, 2010.

[102] AGGARWAL, C. C.. **Outlier ensembles: Position paper**. ACM SIGKDD Explorations, 14:49–58, 2012.

[103] HE, Z.; DENG, S.; XU, X.. **A unified subspace outlier ensemble framework for outlier detection**. Advances in Web Age Information Management, 2005.

[104] KELLER, F.; MULLER, E; BOHM, K.. **Hics: High-contrast subspaces for density-based outlier ranking**. IEEE ICDE Conference, 2012.

[105] LAZAREVIC, A.; KUMAR, V.. **Feature bagging for outlier detection**. ACM KDD Conference, 2005.

[106] LIU, F. T.; TING, K. N.; ZHOU, Z.-H.. **On detecting clustered anomalies using sciforest. machine learning and knowledge discovery in databases**. Springer, p. 274–290, 2010.

[107] MULLER, E.; GUNNEMANN, S.; SEIDL, T.; FARBER, I.. **Tutorial: Discovering multiple clustering solutions grouping objects in different views of the data**. ICDE Conference, 2012.

[108] MULLER, E.; ASSENT, I.; IGLESIAS, P.; MULLE, Y.; BOHM, K.. **Outlier ranking via subspace analysis in multiple views of the data**. ICDM Conference, 2012.

[109] AGGARWAL, C.C.; SATHE, S.. **Theoretical foundations and algorithms for outlier ensembles**. ACM SIGKDD Explorations, 17, 2015.

[110] BARBARA, D.; LI, Y.; COUTO, J.; LIN, J.-L.; JAJODIA, S.. **Bootstrapping a data mining intrusion detection system**. Symposium on Applied Computing, 2003.

[111] MULLER, E.; SCHIFFER, M.; SEIDL, T.. **Statistical selection of relevant subspace projections for outlier ranking**. ICDE Conference, p. 434–445, 2011.

[112] RAYANAAND, S.; AKOGLU, L.. **Less is more: Building selective anomaly ensembles**. ACM Transactions on Knowledge Disovery and Data Mining, 10:42, 2016.

[113] GUHA, S.; MISHRA, N.; ROY, G.; SCHRIJVER, O.. **Robust random cut forest based anomaly detection on streams**. ICML Conference, p. 2712–2721, 2016.

[114] ZIMEK, A.; GAUDET, M.; CAMPELLO, R.; SANDER, J.. **Subsampling for efficient and effective unsupervised outlier detection ensembles**. KDD Conference, 2013.

[115] NGUYEN, H.; ANG, H.; GOPALAKRISHNAN, V.. **Mining ensembles of heterogeneous detectors on random subspaces**. DASFAA, 2010.

[116] GAO, J.; TAN, P.-N.. **Converting output scores from outlier detection algorithms into probability estimates**. ICDM Conference, 2006.

[117] CHEN, J.; SATHE, S.; AGGARWAL, C.; TURAGA, D.. **Outlier detection with autoencoder ensembles**. SIAM Conference on Data Mining, 2017.

[118] AGGARWAL, C. C.; PROCOPIUC, C.; WOLF, J.; YU, P.; PARK, J.. **Fast algorithms for projected clustering**. ACM SIGMOD Conference, 1999.

[119] NEYMAN, J.; PEARSON, E. S.. **Contributions to the theory of testing statistical hypotheses**. Stat. Res. Mem., 1:1:37, 1936.

[120] LOEVE, M.. **Probability theory**. Graduate Texts in Mathematics, 45:12, 1977.

[127] RAYANA, S.. **Outlier Detection DataSets (ODDS)**, 2018. Acesso em: Abril de 2018.

[128] ASUNCION, A.; NEWMAN, D.. **Center for Machine Learning and Intelligent Systems**, 2018. Acesso em: Abril de 2018.

[129] BAO, ZHANA. **Robust subspace outlier detection in high dimensional space**. Cornell University Library, 2014.