Reduction of False Alerts in Monitoring Security Processes using Supervised and Non-Supervised Data Mining Techniques

Ana Raquel Marques Carregã

Department of Engineering and Management, Instituto Superior Técnico, Universidade de Lisboa

June 2018

Abstract

The large amount of data available nowadays is not necessarily a success factor in organizations. In fact, information overload can result in a loss of productivity and may seriously damage the health of an organization. The ORG organization, whose industry must not be disclosed for confidentiality reasons, has developed a mobile application named appMonitor which provides its store managers with a set of alert messages to increase responsiveness in case of irregularities carried out there.

It has recently been identified that the daily volume of messages, in addition to being significantly high, include messages from activities considered regular, distracting managers with harmless information to the business. This raises the need to develop an intelligent data processing mechanism which filters out false alarms and provides users with information truly representative of irregular activity.

Initially, using an Ishikawa Diagram, the factors that caused the problem were identified, which allowed a considerable reduction in the number of alerts. Then, a density-based clustering technique was used to define the boundary that differentiates the true alerts from false alerts, based on the judgment of a panel of business experts.

To automate the message classification were implemented and compared two different tools: control charts and logistic regression, being selected the former. The final system, operating in real time, allowed not only to reduce the volume of alert messages substantially, but also to ensure that the truly critical alerts were sent.

Keywords: data analysis, alert reduction, clustering, logistic regression, control charts

1. Introduction

Considered as the new oil by Shivon Zilis - an investor and partner of Bloomberg Beta - digital information has become an essential asset for value creation in organizations and, according to David Kenny, Managing Director of Watson Artificial Intelligence Software at IBM, may even become a commercial currency (Zilis & Kenny, 2016). Saul Judah, Gartner's Research Director, said information-based technologies are expanding the value of traditional products and services by leveraging digital value to drive business growth (Judah, 2016).

Major advances in the development of data collection techniques and storage technologies have enabled the accumulation of large volumes of data. In absolute values, International Data Corporation estimates that the current volume of digital information around the world will reach 18 zettabytes (18 billion terabytes) and in 2025 it will approach 180 zettabytes (Atreyam, 2016). In Key Marketing Trends for 2017, IBM (2016) states that 90% of the data in December 2016 were recorded in the previous two years, although only 0.5% have been analysed, which shows the deviation between the volume of data collected and the understanding that exists of that same data.

The size and complexity of the content in current databases make it virtually impossible to extract information manually. In fact, the contrast between the continued growth of the volume and complexity of the available data compared with the human processing capacity highlights the need of developing tools capable of extracting information from large volumes of data which, if properly extracted and processed, can lead to an increase of knowledge and value in any sector. According to Shmueli et al. (2010), successful businesses are those that effectively use the data for better predictions, decisions, and strategies.

The present project arises from the identification of an excessive volume of alert messages that are being sent to a mobile business process control application, herein named *appMonitor*. Receiving too much messages, without prior processing, is diluting the purpose of the application as a monitoring tool, thus becoming a source of misinformation. This project explores sophisticated data processing techniques that will allow a controlled and intelligent reduction of the volume of data to enhance the quality and usefulness of the information that reaches the end user.

The remainder of this paper is organized as follows. Section 2 presents the ORG problem. Section 3 details the data sample used. Section 4 presents a literature review. Section 5 presents the approach followed, Section 6 shows the results obtained and Section 7 concludes the paper with some final remarks and ideas for future work.

2. The ORG problem

With commercial activity exclusively in Portugal, the ORG organization is headquartered in Lisbon and has offices in Oporto and Coimbra, currently counting with more than 2000 stores. (The activity sector will not be revealed for confidentiality reasons.)

To stay competitive, ORG decided to invest in the development of a mobile application *appMonitor* that presents updated information of several performance indicators and sends alert messages in case of

identifying irregular behavior. This application is a powerful management support tool, which provides each store manager with real-time business monitoring of risky processes or that somehow represent deviations from normality.

ORG has recently identified that most of store managers receive excessive amounts of alert messages daily, many of them false alarms. This problem makes it difficult to identify situations that are more likely to be considered abnormal, distracting managers with harmless information, which results in a great loss of effectiveness in monitoring. Thus, came the proposal for the development of an intelligent data analysis and processing mechanism to be integrated into the *appMonitor* to reduce the volume of alert messages and send only messages representing irregular behaviour. As the ORG CEO said in an internal statement "we want to imbue the intelligence app (...) so that these alerts become truly meaningful for the managers."

3. The sample used

The global data sample contains all the alerts sent to 200 active stores, about 10% of the total establishments, in the 60 days between October 26 and November 24, 2016. The geographical distribution of the sample is similar to the geographical distribution of the universe (for each district it is ensured that the proportion of sample stores belonging to district X is similar to the proportion of stores in the universe belonging to the same district).

Of the 17 different alert types, the four types with the greatest presence are *Product Price Change*, *Repetitive use of Loyalty Card*, *Sale with stock ≤ 0* and *Sale with PVP <0*, responsible for approximately 80% of the total message volume generated in the sampled period. Thus, the current project focuses on these four types of alert, since they are the ones that contribute most to the problem and whose study can have greater impact in reducing the excessive message volume.

Product Price Change – generated whenever the price of a product is modified in the system. Syntax: "The product *product code* has changed from *initial value* to *final value* (*value change*) by operator *operator number* with the following justification: *justification*."

Repetitive use of Loyalty Card – generated whenever, during a day, an operation is carried out with a loyalty card. Syntax: "The operator *operator number* performed *number of operations* operations with card *card number* accumulating *number of points* pts assigned."

Sale with stock ≤ 0 – generated whenever there is a sale of a product whose availability in the system is negative or zero. Syntax: "The operator *operator number* has made the sale *sales number* with *number of units* unit (s) of the product *product code* with the value *value* without stock available and without registration of a customer reservation."

Sale with PVP <0 – this alert is issued whenever, in a sales process, there is one or more products sold with negative PVP. Syntax: "The operator *operator code* made the sale *sale number* with *U* units of product *product code* with negative PVP (*PVP value*)."

4. Literature review

4.1. Similar Problems

The false-alerts reduction plays a very important role in the quality of the information obtained, and the study of this problem has been reinforced especially in areas where the excess of false positives is truly critical, particularly in intrusion detection systems. There have been significant efforts in developing complementary data post-processing mechanisms reduce the volume of false positives, which makes this application area considerably similar to the problem under study.

4.1.1. Intrusion Detection Systems

Intrusion Detection Systems (IDS) (Anderson, 1980) are algorithms for monitoring computer systems that seek to detect unauthorized activities that may compromise the confidentiality, integrity, and availability of a particular environment. Despite the developments in this area it has been observed that these systems generate thousands of alerts per day, which in most cases, are false positives. Julisch (2001) has developed the concept of *alarm clustering* for IBM, in which alerts should be addressed by identifying their root causes, assuming that similar alerts have common causes. All the alerts that meet some criteria of similarity should be grouped in the same cluster so that, when identifying the root cause of one alert, it is possible to explain the origin of all the alerts belonging to that same cluster. Viinika and Debar (2004) propose the implementation of a variant of Exponentially Weighted Moving Average (EWMA) charts, in order to monitor the number of alerts generated by the IDS. The analysis of alerts is done by comparing the message flow with a reference value. If the alert flow exceeds the reference value, then the system classifies as irregular all alerts generated during that period. Pietraszek (2004) presents the Adaptive Learner for Alert Classification tool (ALAC). The algorithm labels each message as true or false alert and adjusts its parameters through feedback from an analyst who verifies that the message has been correctly classified. Pietraszek and Tanner (2005) suggest the implementation of two complementary approaches. First, they use the concept of alarm clustering, presented by Julisch (2001), which they call Clustering Alerts for Root Causes (CLARAty). Then, for the alerts that were considered true positives in the first phase, they apply the ALAC tool. Tjhai et al. (2010) use a combination of two data mining techniques to filter out false positives in large volumes of alerts. Firstly, a Self-Organizing Map (SOM) (Kohonen, 1997) produces a map of the input data according to a certain degree of similarity. The data is then analyzed and later separated into distinct categories, functioning as a compression tool. The categories mapped by the SOM tool are then classified as false or true positives using the K-means clustering method (Macqueen, 1967).

4.2. Clustering Techniques

Clustering or data segmentation is the process of grouping sets of objects into classes based on similarity criteria (Matheus et al., 1993). According to Han and Kamber (2016), it is possible to group the methods into four distinct classes: partitioning, hierarchical, density-based and grid-based.

4.2.1. Partitioning methods

Given a set of *n* observations, a partitioning algorithm organizes the data into *k* partitions or clusters. Clusters are formed in order to optimize a partitioning criterion that typically involves maximizing the similarity between observations in the same cluster and maximizing dissimilarities between different clusters, simultaneously.

One of the most used algorithms is K-Means (Macqueen, 1967), where the degree of similarity is measured based on the average distance between the data belonging to the same cluster (distance to the centroid). According to Celebi and Kingravi (2015), this method has good results, especially when the data structure is composed by compact clouds clearly separated from each other. However, this method requires the definition of the number of clusters to be formed, thus requiring prior knowledge of the structure and content of the data (Jin & Han, 2010). The fact that it involves distance calculations is another limitation of this method, especially when non-numeric attributes are involved. According to Kumar et al. (2017) it is very sensitive to noise and outliers: a reduced amount of data can substantially influence the location of the centroids and consequently the composition of the clusters.

Other commonly used methods are K-Medoids (Kaufman & Rousseeuw, 1987), Clustering Large Applications (CLARA) (Kaufman & Rousseeuw, 1990) and Clustering Applications Large Applications Based on Randomized Search (CLARANS) (Ng & Han, 1994).

4.2.2. Hierarchical methods

A hierarchical algorithm groups the data into clusters of distinct levels to build a cluster tree, or dendrogram, such that clusters of a given level may be joined together to form a cluster at the following level, according to a similarity criterion. The number of clusters is defined by selecting the minimum degree of similarity between observations of the same cluster. A reduced degree of similarity allows objects with less similarities to each other to belong to the same cluster. On the other hand, imposing a higher degree of similarity between observations of the same cluster, orange line, will result in a higher number of clusters. One of the advantages of hierarchical clustering is that the dendrogram can be cut into different levels to obtain different clustering solutions for the same data set, without having to perform the algorithm again. However, these models are irreversible: once the algorithms are performed, they cannot be reverted (Berkhin, 2006).

4.2.3. Density-based methods

The first approach of density-based algorithm arose in 1996 due to the inexistence of computationally efficient solutions for discovering clusters with arbitrary shape without previous knowledge of the data structure. Ester et al. (1996) present the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. The authors point out that the main reason for identifying the clusters shown in Figure 1 is due to the fact that each of them has a typical density of points, considerably higher inside the clusters when compared to the the areas of noise.

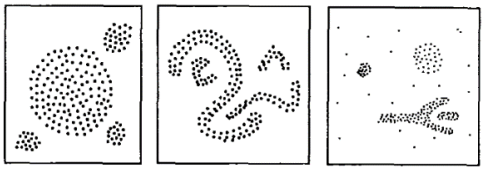


Figure 1 – Pattern recognition from density variations (Ester et al., 1996)

The algorithm analyses each observation individually and calculates its neighbourhood according to a predefined radius. For an observation to be included in a cluster, its neighbourhood must contain at least a minimum number of points, that is, the density of neighbour points should surpass a predefined boundary. In this way the algorithm can efficiently find clusters arbitrarily and distinguish them from surrounding noise.

Among other algorithms developed for the same purpose are DENCLUE (Hinneburg & Keim, 1998) and OPTICS (Ankerst et al., 1999).

4.2.4. Grid-based methods

Grid-based methods divide the data space into a finite number of grid-structured cells, under which all clustering operations are performed. The data are distributed by the cells according to certain criteria of similarity and after that, for each cell individually, some statistical operations are made using the points inside. Clustering operations are then performed on each of the cells, rather than being applied to all data objects individually. Since the number of cells is typically much lower than the data volume, processing becomes significantly more efficient.

Other algorithms in the same category are STING (Wang et al., 1997), Wave-Cluster (Sheikholeslami et al., 2000) and CLIQUE (Agrawal et al., 2005).

4.3. Control Charts

Control charts are statistical tools for process control that use time evolution plots of quality parameters. Each plot contains a middle line, which represents the mean value of the quality parameter, and two horizontal lines for the upper and lower control limits (see Figure 2). Limits are defined from the statistical analysis of historical data, and delimit the controlled range of the parameter, characterized by a stable pattern associated with common causes. An out-of-bound value evidences that the process is out of control, requiring investigation and corrective action to find and eliminate its causes (Montgomery, 2009).

Control charts can be classified into variable charts or attribute charts. A quality parameter represented by a value on a continuous scale is called a variable and its control should be carried out through control charts that analyse the mean and standard deviation, called variable charts. Quality parameters that cannot be measured on a continuous scale are designated as attributes and are studied through attribute control charts. One of the types of analysis allowed by these charts is the monitoring of the ratio of nonconformities (for example, the percentage of defective parts in a sample) via the nonconformity fraction control chart, also referred to as *p* chart. In other circumstances, it may be more convenient to study the absolute volume of defects or nonconformities observed, where a nonconformity control chart or Poisson chart is used.

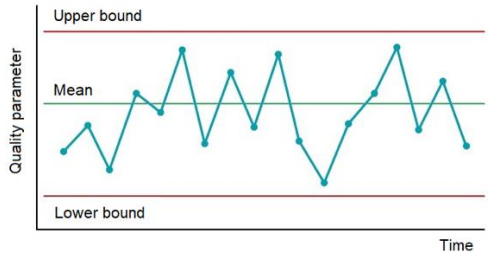


Figure 2 – Representation of a control chart, adapted from Montgomery (2009)

4.4. Regression Techniques

Regression models are one of the most important techniques in statistical analysis of data when modelling the relationship between variables (Hosmer & Lemeshow, 2000). The main objective of these models is to explore the relationship between one or more explanatory variables (independent variables) and a response variable (dependent variable).

One of the most common examples of regression is the linear regression model, where it is assumed that the output variable is continuous. In cases where the response variable is dichotomic, the logistic regression model is the most popular.

4.4.1. Logistic Regression

Logistic regression is a statistical technique whose objective is to model the nonlinear relationship between a dichotomic response variable and one or a set of explanatory variables, from a set of observations. The logistic regression model is univariate in the case where there is a single independent variable and multivariate when there is a set of *p* independent variables.

According to Kutner et. al, (2005) any regression problem aims to obtain an estimate of the expected value of a response variable Y given an independent variable x, $E[Y|x]$. In the case of linear regression, it is assumed that the mean value (1) can be expressed as a linear function of x and may take any value between $-\infty$ and $+\infty$.However, in context of binary response, the expected value of the response variable must belong to the interval $[0,1]$ (2).

$$\mathrm{E[Y|x]} = \beta_0 + \beta_1 \mathrm{x} \qquad (1)$$

$$0 \leq E[Y|x] \leq 1 \qquad (2)$$

Thus, the variation of the mean value per unit of variation of x should become progressively smaller as $E[Y|x]$ approaches 0 or 1 (gradual approximation to the extreme values), obtaining a S-shaped curve.

For the curve modelling, it is used the logistic distribution function, from which it follows that the response variable Y is a Bernoulli random variable, that takes values between 1 or 0 with the probability $\pi(x)$ or $1-\pi(x)$, respectively, and whose expected value is given by (3).

$$E[Y|x] = \pi(x) = \beta_0 + \beta_1 x \qquad (3)$$

The response function of the logistic regression is given by the expected value of Y given x (4), whose parameters are estimated by the maximum likelihood method given a set of observations.

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \qquad (4)$$

The *logit* transformation (5) is applied, from which it results (6).

$$g(x) = ln\left[\frac{\pi(x)}{1-\pi(x)}\right] \qquad (5)$$

$$g(x) = \beta_0 + \beta_1 x \qquad (6)$$

The importance of the *logit* transformation is because it is linear in its parameters, can be continuous and can take values between -∞ and +∞, as linear regression.

Once the response variable is dichotomous, it must be described as a function of the independent variable (7), where ε is the response error: if $y=1$, $\varepsilon = 1-\pi(x)$ with probability $\pi(x)$, while for $y=0$, $\varepsilon = -\pi(x)$ with probability $1-\pi(x)$.

$$y = \pi(x) + \varepsilon \qquad (7)$$

Thus, ε takes a distribution with mean value 0 and variance $\pi(x)[1-\pi(x)]$.

5. The Research Approach

Before proceeding to data processing, it was performed a detailed analysis of the data in order to identify possible causes for the presence of excessive volumes of false alerts. An Ishikawa diagram, also called a cause-and-effect diagram, has been built to investigate the causes of the problem. The implementation and results are presented in section 6.1. The remaining messages, whose cause remains unidentified, indicate the need to use other types of techniques.

One of the major challenges underlying the present work is the fact that there is no well-defined boundary between true and false alerts. For this reason, it was necessary to develop a classification model of alert messages based on the judgment of a panel of experts, business specialists, to guarantee supervision.

Firstly, the attributes to be presented to the model were selected, and then their graphics were presented to the panel of experts, which enabled the construction of a set of criteria for the message classification for the distinct types of alerts.

The clustering technique DBSCAN was applied for data processing and identification of true positives (see section 6.2), to serve as a reference for the message classification model developed in section 6.3.

Noting the importance of not exploring a single classification tool, two different approaches were implemented and compared: control charts and logistic regression. The first one is already known by decision makers, which facilitates the interpretation of the results, does not require significant computational efforts and can be used in real time; the logistic regression allows the development of binary classification models from attributes with arbitrary distribution without needing previous information about their statistical behaviour.

The comparison of the models is carried out using classification tables, from which performance is measured. Four indicators frequently used in classification problems (Olson & Delen, 2008) are calculated: sensitivity, specificity, precision and accuracy.

To summarize, the DBSCAN will be applied to the training sample, and will classify all the data. This classification will be the learning base for logistic regression models and control charts. The both models to be developed will then have a learning phase, or training phase, in which they will analyse the

data set previously classified by the DBSCAN. Then the evaluation phase of the model performance, or test phase, will be carried out by analysing the classification of an unknown data set, called the test sample. The training sample contains the messages generated in the 30 days between 28/09/2016 and 25/10/2016 (training period), while the test sample is composed of the messages generated in the 30 days between 26/10/2016 and 24/11/2016 (test period).

6. Results

6.1. Ishikawa Diagram

It was constructed a single Ishikawa diagram (see Figure 3) since the identified causes are transversal to all types of alerts. The set of causes was grouped into three categories: ***Algorithm***, ***App Structure*** and ***Operators***.

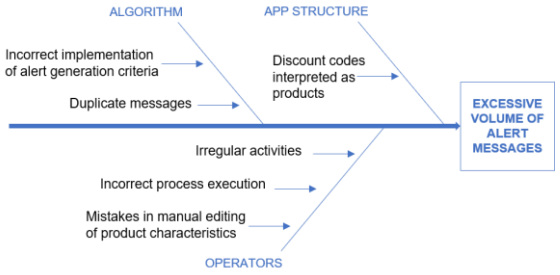


Figure 3 – Ishikawa Diagram

Algorithm: covers the causes related to the inadequate technical or functional implementation of the prerequisites defined for the appMonitor. The existence of computer gaps that generate duplicate messages is a potential cause of the problem. Another cause is the incorrect application of the alert generation criteria, causing the message filtering process to be applied with incorrect reference values.

App Structure: causes related to the appMonitor architecture. One of the factors that may be a root-cause of the problem concerns product codes: it has been found that each store creates its own discount codes, and these are interpreted by the system as product codes. Thus, appMonitor current algorithm, when analysing sales data, is not only evaluating the prices and quantities of products sold, but also the number of discount codes applied on each sale, and their discounted values.

Operators: includes the causes directly related to the operators' activities. It is part of the appMonitor's objective to select the alerts that are representative of irregular or fraudulent activities, or incorrect implementation of processes which will always be associated with the operators' activities. Therefore, it is desired that the respective messages continue to be sent to the store managers.

Figure 4 shows the results for the **Sale with stock ≤ 0** alert type. From the initial set of 412,370 messages, it was found that 203,449 (49.34%) are generated due to price change of discount codes, and that 45,117 (10.94%) are duplicate messages. The correct implementation of the message generation criteria was verified, since all generated alerts relate to sales of products with stock values equal to or less than zero. After cause removal there was a reduction of 60.28% in the volume of messages for this type of alert.

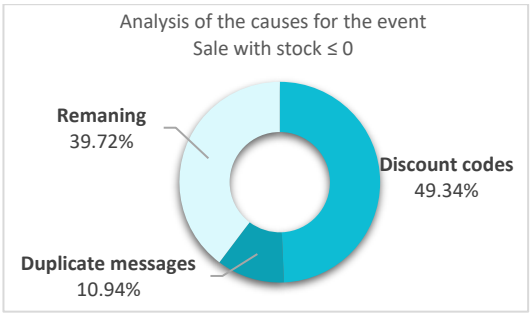


Figure 4 - Segmentation of *Sale with stock ≤ 0* messages for identified causes.

To obtain an impact overview of the identification and removal of the causes mentioned above, it follows Figure 5, which summarizes the percentage reduction in alert volume.

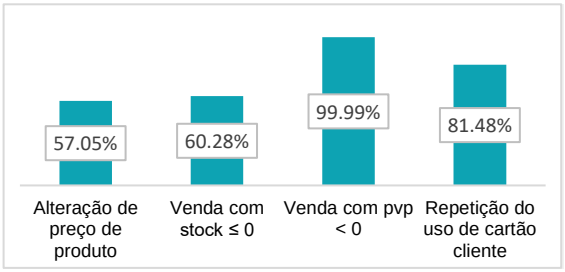


Figure 5 – Percentage reduction in the overall message volume resulting from the removal of all identified causes, by alert type.

Regarding the *Product Price Change* alert, it was possible to eliminate 57.05% of messages. The impact was similar for the *Sale with stock ≤ 0* alert, with a 56.29% reduction in message volume. The *Sale with PVP < 0* alert suffered the greatest reduction, where 99.99% of its messages were identified as false positives. Lastly, *Repetitive use of Loyalty Card* messages suffered an overall reduction of 81.48%.

The remaining messages, whose cause remains unidentified, point to the need to resort to other types of tools, which will be explored in the following sections.

6.2. Modelling the expert's knowledge using the density-based technique DBSCAN

One of the aspects considered for the attributes selection was the trade-off between the amount of information and the comprehensibility of the results. It was chosen a two-dimensional analysis, counting on the two most distinctive attributes of each type of alert (see Table 1).

Table 1 –Selected attributes for each alert type

Repetitive use of Loyalty Card
- Number of operations
- Number of transacted points
Product Price Change
- Initial value (€)
- Final value (€)
Sale with stock ≤ 0
- Number of units
- Line value (€)

To create a general rule for each type of alert to be applied to each store individually, it was conducted an analysis with data from all the stores of the sample. Figure 6 shows the distribution of the attributes selected for the *Sale with stock ≤ 0* alert type,

complemented by Table 2, which lists the minimum value, maximum value and quartiles.

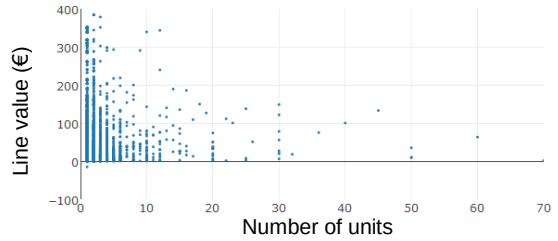


Figure 6 – Attributes for the alert type *Sale with stock ≤ 0* (Number of units and Line value). Set with 163 804 messages.

Each point in Figure 6 refers to a sale in which it was sold at least one product with stock ≤ 0. The number of units in these conditions varied between 1 and 72 per sale, and in at least 75% of the cases the number of units was equal to one. The line value, that is, the value of the products sold in these conditions varies between -15€ and 19,999.98€, with 50% of the messages assuming values between 4.92€ and 18.35€ per sale.

Table 2 – Minimum, maximum, and quartiles of the attributes Number of units and Line value for the alert *Sale with stock≤0*

Attribute	*Min.*	Q_1	Q_2	Q_3	*Max.*
Number of units	1	1	1	1	72
Line value (€)	-15	4.92	8.94	18.35	19,999.98

Experts panel observations: The higher the number of units sold for a given product without stock, the more important it becomes to send the alert to the store manager. It is practically inevitable the existence of small stock differences. According to the experts, a difference of one or two units is acceptable.

Regarding the sales value of products with negative stock, at least 50% of the alerts are in an acceptable range of values (between 4.92€ and 18.35€) in line with the average sales prices applied to the final consumer. The higher the value, the more distant one is from the values usually practiced in the stores, and therefore, more attention will be need from the managers.

From the experts' observations it is possible to extract the basic notion that, for each store, the greater the number of points in the same area of the graph (higher density), the more representative are those same points of the *modus operandi* of the respective store. In other words, the smaller the density of points, the less representative they are of the regular behaviour of the store, requiring more attention from the managers. Although the analysis was carried out with the data of all the stores simultaneously, it was considered acceptable to assume that it remains valid for each one individually.

It was applied the DBSCAN clustering method (Ester et al., 1996) due its ability to cluster points based on density. Points inside the cluster with higher density are considered as false positives. All the remaining points will be considered outliers and should therefore be sent to the store manager. Figure 7 shows the results of the DBSCAN application to the training sample of one store (pilot store), with the points classified as true alerts in red and false alerts in blue.

These results are in line with the experts' comments: alerts referring to a sale of one unit of a specific product without stock with average sale amount were classified as false alerts. Messages associated with high sales amounts (valuable products), although corresponding to the sale of a single product, were classified as outliers.

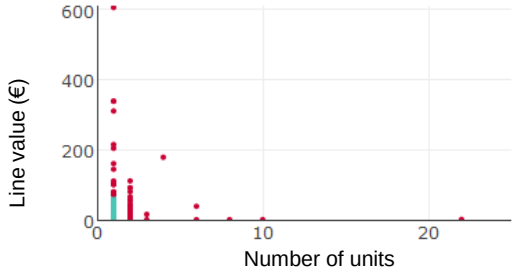


Figure 7 – *Sale with stock ≤ 0*: result of DBSCAN application to the pilot store training sample. Boundary between regular (blue) and irregular (red) points.

During the training period the pilot store received 587 alerts. The DBSCAN allowed to filter 511 messages, which were classified as false positives, the remaining 76 were considered outliers. This is equivalent to a reduction of 87.05%.

The application of DBSCAN had a significant impact on reducing the message volume from the 200 stores: in 152 stores, message reduction was between 80% and 100%, 41 stores had a reduction of between 60% and 80%. that only seven stores suffered a reduction of less than 60%.

6.3. Choosing a classifier: Logistic Regression or Control Charts

Control charts

Once the selected attributes are all numerical, only average control charts will be used - one for each attribute, that is, two for each type of alert referred in Table 1. The reference values are calculated with the messages previously classified by the DBSCAN as false alerts (on other words, messages originated from regular causes), in order to determine the upper control limits - see equations (8), (9) and (10).

$$LSC = \bar{\bar{X}} + 2\sigma_{\bar{X}} \quad (8)$$

$$\bar{\bar{X}} = \frac{\sum_{i=1}^{30} \bar{X}_i}{30} \quad (9)$$

$$\sigma_{\bar{X}} = \sqrt{\frac{\sum_{1}^{30} \left(\bar{\bar{X}} - \bar{X}_i\right)^2}{30}} \quad (10)$$

In the test phase, all messages from the test sample previously classified by the DBSCAN will be analysed by the control charts. Messages whose attribute values exceed the defined limits will be classified as positive and the remaining messages as negative.

Logistic Regression

A multivariate model was developed, with two input variables referring to the two attributes selected for each type of alert (see Table 1). Unlike control charts, logistic regression will have a training phase with all points of the training sample.

The regression model was developed using the *glm* function, whose implementation was carried out in language R integrally. This will be analysed from the values of the conditional probability P(Y=1|X), with the

decision limit of 0.5. Thus, the model will classify as positive a message with an associated conditional probability P(Y=1|X)>0.5 and as negative in the opposite case.

The two models will be evaluated through their respective classification tables (Table 3). The results will be compared with the results of the DBSCAN classification.

Table 3 – Classification table template, adapted from (Olson & Delen, 2008)

		Model classification result	
		0	1
DBSCAN	1	False Negatives (FN)	True Positives (VP)
	0	True Negatives (VN)	False Positives (FP)

The performance of both models will be measured through four indicators often used in classification problems (Olson & Delen, 2008): **sensitivity** (11), **specificity** (12), **precision** (13) and **accuracy** (14).

$$\text{Sensibility} = \frac{VP}{VP + FN} \quad (11)$$

$$\text{Specificity} = \frac{VN}{VN + FP} \quad (12)$$

$$\text{Precision} = \frac{VP}{VP + FP} \quad (13)$$

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + VN + FN} \quad (14)$$

Results for the pilot store

The results of the two models for the test sample of the pilot store, with messages of type *Sale with stock ≤ 0* are presented below. Table 4 and Table 5 refer to the results obtained by control charts and logistic regression models, respectively.

Table 4 – Control chart classification table for the *Sale with stock ≤ 0* alert.

		Control charts	
		0	1
DBSCAN	1	0	55
	0	406	48

The test sample consists of 509 messages, of which 55 are true alerts. The control charts model correctly classified all true alerts, which corresponds to a sensitivity of 100%, and for the 454 false alerts, 406 were correctly classified, which is equivalent to a specificity of 85%. The model has a precision of 53% given that of the 103 messages it rated positive, only 55 are real positive, and 91% accuracy for having correctly classified 461 messages from the entire sample.

Table 5 – Logistic regression classification table for the *Sale with stock ≤ 0* alert.

		Logistic regression	
		0	1
DBSCAN	1	15	40
	0	454	0

The logistic regression model correctly classified 40 of the 55 true alerts (sensitivity of 73%) and all outliers (100% specificity). It achieved a precision of 100% since all the messages it identified as positive were actually true alerts. Finally, of the 509 messages in the sample, 494 were classified correctly, which corresponds to an accuracy of 97%. The performance indicators of both models are summarized in Table 6.

Table 6 – Comparison of the metrics resulting from the application of control charts and logistic regression, for the **Sale with stock ≤ 0** alert.

Indicator	*Control charts*	*Logistics regression*
Sensibility	1.00	0.73
Specificity	0.89	1.00
Precision	0.53	1.00
Accuracy	0.91	0.97

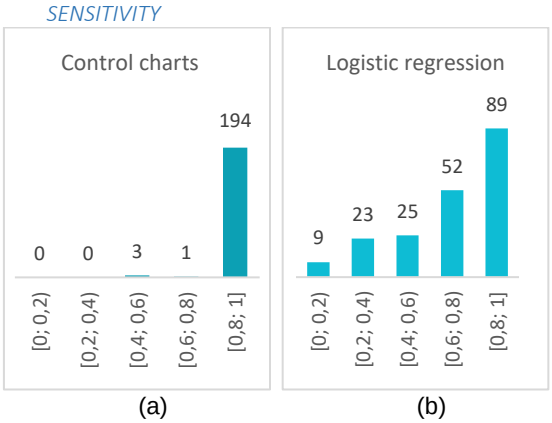
The control charts correctly identified all real alerts, performing better than logistic regression, which identified only 73%. Regarding the specificity, the logistic regression obtained a better performance by having correctly identified the totality of the outliers, compared to the 89% obtained by the chart of control charts. In terms of precision, the logistic regression model stood out with 100% compared to the control cards model, in which only 53% of the points that classified as positive were effectively. The models do not differ significantly in accuracy, having achieved results above 90%.

The control charts had superior performance in the identification of the true alerts, whereas the logistic regression obtained superior results in the classification of outliers.

Results for the stores of the sample

The results for all the stores in the sample are shown, with the histograms of the four indicators, resulting from the application of control charts and logistic regression models to the *Sale with stock ≤ 0* messages. Figure 8 is related to the sensitivity and Figure 9 shows the specificity. The results of the precision indicator are shown in the histograms of Figure 10, while those for accuracy are shown in Figure 11.

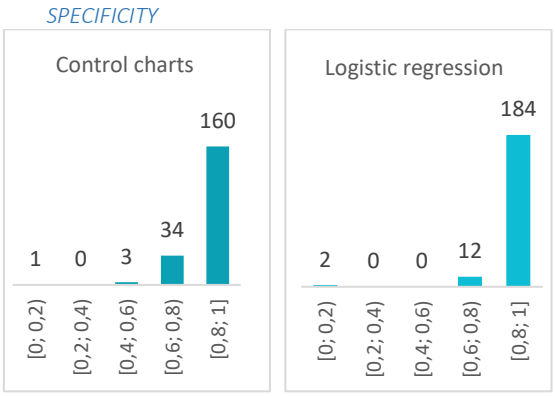
The difference in sensitivity values is significant, with the control charts model being the most successful in classifying true alerts. Of the 198 stores analysed, it was possible to correctly identify between 80% and 100% of the actual alerts of 194 stores, against the 89 stores using the logistic regression model. The results of the logistic regression model are more dispersed, including 9 stores whose sensitivity did not exceed 20%, indicating that for these stores less than 20% of real warnings would be sent.



(a) (b)

Figure 8 – Histogram of Sensitivity, in the sample of the 200 stores, for the *Sale with stock ≤ 0* alert, resulting from the application of: (a) control charts (b) logistic regression.

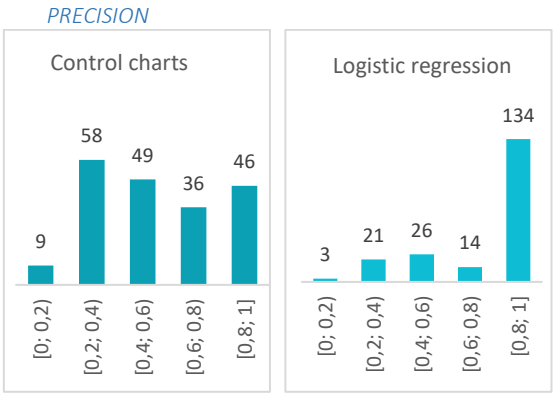
The logistic regression model has obtained better results, once it has identified between 80% and 100% of the false alerts of 184 stores, while the control charts model has achieved similar results for 160 stores. The distribution of results with the control charts model is slightly more dispersed, with more stores having smaller specificity values, that is, with less ability to correctly classify false alerts.



(a) (b)

Figure 9 – Histogram of Specificity, in the sample of the 200 stores, for the *Sale with stock ≤ 0* alert, resulting from the application of: (a) control charts (b) logistic regression.

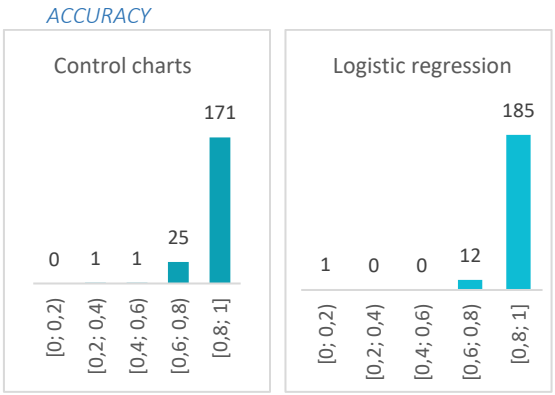
The logistic regression model was able to ensure that, for 134 stores, 80 to 100% of the messages were correctly classified, in contrast to the 46 stores in the case of the control charts model. In 14 stores, the precision reached values between 60 and 80%, but the remaining 50 stores reached lower values, between 0 and 60%. The distribution of the results of the control charts model is noticeably more dispersed, including 67 stores whose precision did not reach 40%, denoting that at least 60% of messages classified as positive were false positives.



(a) (b)

Figure 10 – Histogram of Precision, in the sample of the 200 stores, for the *Sale with stock ≤ 0* alert, resulting from the application of: (a) control charts (b) logistic regression.

Regarding the accuracy indicator, both models obtained favourable results, showing a slight superiority in the results of the logistic regression model. It was able to correctly classify between 80% and 100% of messages from 185 stores in a universe of 198, while the control charts model reached similar values in 171 stores.



(a) (b)

Figure 11 – Histogram of Accuracy, in the sample of the 200 stores, for the *Sale with stock ≤ 0* alert, resulting from the application of: (a) control charts (b) logistic regression.

Method Selection

Regarding the **sensitivity** indicator, the results of the control charts model were globally higher than those obtained by the logistic regression model in the three types of alerts studied. This means that this model was more successful at correctly classifying the true positives, ensuring the sending of higher percentages of true alerts. This feature is of the greatest importance since store managers are expected to receive as many true alerts as possible.

Regarding the **specificity** indicator, the results obtained by the logistic regression model were better, except for the type of alert *Repetitive use of Loyalty Card*, in which the control chart model stood out positively. The greater the specificity, the better the ability to correctly classify false positives, discharging the end user from receiving messages that do not represent any level of irregularity.

The results obtained for the **precision** indicator with the logistic regression model were significantly higher for the *Product Price Change* and *Sale with Stock ≤ 0* alerts, while for the *Repetitive use of Loyalty Card* alert the control charts model obtained slightly better results. The higher the value of this indicator, the higher the proportion of alerts correctly identified as positive versus all positively ranked, which translates into the quality of the volume of alerts sent through the appMonitor.

Regarding the **accuracy** indicator, the application of the two models achieved similar results, except for the type of alert *Repetitive use of Loyalty Card*, where the control charts model has shown a better performance in the message classification. The greater the accuracy of a model, the greater the ability to correctly classify the overall messages.

It should be reminded that, on the one hand, the purpose of the appMonitor is to send alerts to its end user and, on the other hand, the present project arose from the need to reduce the volume of false alerts that are being sent. It is intended that the classification model correctly identifies the highest volume of alerts possible, to be able to send the highest percentage of true alerts and omit the highest percentage of false positives. From the point of view of the analysed indicators, this means a combination of high sensitivity with high specificity, which results in high values of accuracy.

In the whole sample of the 200 stores, the control charts model achieved the best results for the sensitivity while the logistic regression model was more successful with respect to the specificity. Thus, to select one model, it was necessary to evaluate and compare the importance of better classifying the true alerts and better identifying the false positives, that is, between sending a higher percentage of true positives and omitting the greater number of false positives. For such, it is important to emphasize the importance of sending the highest possible percentage of true alerts previously classified by the DBSCAN clustering method.

At this point, it is evident the importance of ensuring that the highest level of true alerts is sent to the store managers, even if it implies the sending of some false positives. Therefore, the control charts model was selected since it showed better performance in the classification of true alerts.

Final Results

After the selection of the control charts model, the results of the application of the final system to the test sample with the 200 stores are presented below. The results presented refer exclusively to the test sample given that the training sample was only used to the learning phase of the classification model.

The results for the *Sale with stock ≤ 0* alert type are shown in Table 7: the Total column indicates the total number of messages from all stores in each process phase, while the remaining columns show the dispersion of the number of messages between stores. Figure 12 presents the result of the distinct process phases as a percentage of the initial message volume of all stores.

Table 7 - Results of the final system implementation, with total number of messages and dispersion between stores, for *Sale with stock≤0* alert

	Total	*Min*	*Q_1*	*Q_2*	*Q_3*	*Max*
Initial sample	211,931	59	326	549	1056	9297
Sample after cause removal	83,099	41	184	300	449	7949
Final sample	16,920	8	42	62	98	1175

Of the 211,931 messages for the test period of all the stores of the sample, it was possible to identify the causes and remove 61% of the initial quantity. The implementation of the control charts model to the remaining messages also allowed a further 31% reduction of the initial volume, consisting of false alerts. Only 8% of the initial message sample were classified as critical for the stores.

Regarding the individual store results, it can be observed that the initial volume of alerts, ranging from 59 to 9297 messages (see line 1 of Table 7), ranged from 41 to 7949 messages after identification and removal of causes (see line 2 of Table 7). The implementation of the second phase of the process was equally successful, resulting in an even greater reduction in the quantity of alerts. In this final sample (see line 3 of Table 7), the number of alerts varies between 8 and 1175 messages, with at least 75% of the stores receiving fewer than a hundred alerts in the 30 days of the test, which is a considerable reduction in the number of messages sent to the end user.

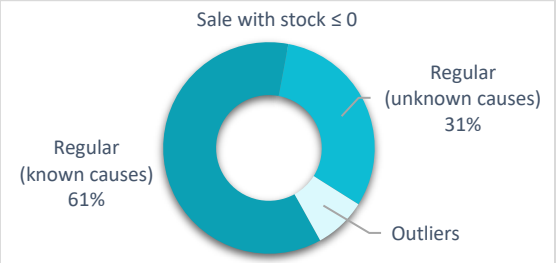


Figure 12 - Result of the process steps as a percentage of the initial volume of all stores, for the *Sale with stock ≤ 0* alert.

For the remaining types of alerts analysed, the result was similar, with a significant reduction in the volume of alerts sent to stores and in the dispersion of messages.

7. Final remarks and future work

The final system, designed to operate in real time, allowed a significant reduction in the volume of messages to be sent to store managers, while ensuring the selection of truly critical and worthy alerts. The development of a robust system was only possible due to the fact than some details were constantly taken into account, such as the business characteristics, restrictions on the integration of new developments in the application, computational efficiency (since it was intended a classifier with real-time responsiveness), ease of interpretation of the results and relative simplicity of the tools (in order to reduce the "black box" effect). These aspects allowed to create a system adapted to the needs of the ORG, which fulfilled the proposed objectives.

This project has given the visibility to the stakeholders of content of the alert messages, which had not been occurred so far. In fact, at the beginning of the analysis, it was observed a total lack of knowledge about the content of what was being generated by the appMonitor. In addition, new tools for data analysis and pattern detection have been explored, allowing ORG not only to improve the appMonitor, but also the opportunity to present these tools to other departments, studying the feasibility of applying them to other internal projects.

However, it should be remembered that four types of alert were analyzed, out of a total of 17. In the future it may be appropriate to analyze the remaining types and verify the viability of the implementation of the same tools. Such an analysis may enable the identification of new causes, increasing the likelihood of further reducing the volume of false alerts. It may also be interesting to extend the message analysis to more than two attributes; it is believed that including in the model information about the operator who generated the alert could be potentially significant for the improvement of the classification system.

The system was developed in order to apply the same tools to all ORG stores. But knowing in advance that there are differences between stores regarding to geographical location, sales volume, number of employees, operation hours and marketing strategy, they may have completely dissimilar modes of operation. Thus, it may be fruitful to analyse them separately in order to find characteristics that allow them to be assigned a classification and group them into separate sets. In this way, a store-oriented classification system can be developed, applying

different tools to distinct types of stores, seeking to adjust to the business characteristics and then to improve the services provided by ORG.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (2005). Automatic Subspace Clustering of High Dimensional Data. *Data Mining and Knowledge Discovery*, *11*(1), 5–33.

Anderson, J.P. (1980). *Computer security threat monitoring and surveillance*. *Technical Report*. Fort Washington, PA: James P. Anderson Co.

Ankerst, M., Breunig, M. M., Kriegel, H., & Sander, J. (1999). OPTICS: Ordering Points To Identify the Clustering Structure. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data - SIGMOD '99* (pp. 49–60).

Atreyam, S. (2016). *The Internet of Things is Blowing up Everything*. International Data Corporation.

Berkhin, P. (2006). Survey of Clustering Data Mining Techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping Multidimensional Data: Recent Advances in Clustering* (pp. 25–71). Springer Berlin Heidelberg.

Celebi, M.E., & Kingravi, H. A. (2015). Linear, Deterministic, and Order-Invariant Initialization Methods for the K-Means Clustering Algorithm. In M. E. Celebi (Ed.), *Partitional Clustering Algorithms* (pp. 79–98). Springer International.

Ester, M., Kriegel, H.P., Sander, J., & Xu, X. (1996). A Density Based Notion of Clusters in Large Spatial Databases with Noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*.

Han, J., & Kamber, M. (2016). *Data Mining - Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.

Hinneburg, A., & Keim, D.A. (1998). DENCLUE : An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 58–65).

Hosmer, D.W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd Ed.). John Wiley & Sons.

IBM. (2016). 10 Key Marketing Trends for 2017. Retrieved from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN

Jin, X., & Han, J. (2010). Partitional Clustering. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 766–766). Boston: Springer.

Judah, S. (2016). Disruption and the Digital Value of Business - The Rise of Digital Economics. Retrieved from https://www.gartner.com/doc/3456117/disruption-digital-value-business-

Julisch, K. (2001). Mining Alarm Clusters to Improve Alarm Handling Efficiency. In *ACSAC '01. Annual Computer Security Applications Conference* (pp. 12–21). New Orleans.

Kaufman, L., & Rousseeuw, P.J. (1990). *Finding Groups in Data - An Introduction to Cluster Analysis*. Hoboken, New Jersey: John Wiley & Sons.

Kaufman, & Rousseeuw. (1987). *Clustering by Means of Medoids*. *Statistical Data Analysis Based on the L 1-Norm and Related Methods. First International Conference*.

Kohonen, T. (1997). *Self-Organizing Maps*. Berlin: Springer.

Kumar, R., Lu, K., Ave, N.G., Moseley, B., Drive, B., & Louis, S. (2017). Local Search Methods for k-Means with Outliers. In *Proceedings in the Very Large Data Bases Endowment* (Vol. 10, pp. 1–12).

Kutner, M.H., Nachtseim, C., & Neter, J. (2005). Applied Linear Regression Models. *Applied Linear Regression Models*.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*(233), 281–297.

Matheus, C.J., Chan, P.K., & Piatetsky-Shapiro, G. (1993). Systems for knowledge discovery in databases. *IEEE Transactions on Knowledge and Data Engineering*, *5*(6), 903–913.

Montgomery, D.C. (2009). *Introduction to Statistical Quality Control*. USA: John Wiley & Sons.

Ng, R.T., & Han, J. (1994). Efficient and Effective Clustering Data Mining Methods for Spatial. In *Proceedings of 20th International Conference on Very Large Data Bases* (pp. 144–155). Santiago, Chile.

Olson, D.L., & Delen, D. (2008). *Advanced Data Mining Techniques*. Berlin: Springer-Verlag.

Pietraszek, T. (2004). Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In E. Jonsson, A. Valdes, & M. Almgren (Eds.), *Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15 - 17, 2004. Proceedings* (pp. 102–124). Berlin: Springer.

Pietraszek, T., & Tanner, A. (2005). Data mining and machine learning—Towards reducing false positives in intrusion detection. *Information Security Technical Report*, *10*(3), 169–183.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (2000). WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, *8*(3), 289–304.

Shmueli, G., Palet, N.R., & Bruce, P.C. (2010). *Data Mining for Business Intelligence*. New Jersey, USA: John Wiley & Sons.

Tjhai, G.C., Furnell, S.M., Papadaki, M., & Clarke, N. L. (2010). A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. *Computers and Security*, *29*(6), 712–723.

Viinikka, J., & Debar, H. (2004). Monitoring IDS background noise using EWMA control charts and alert information, *3224*, 166–187.

Wang, W., Yang, J., & Muntz, R. (1997). STING : A Statistical Information Grid Approach to Spatial Data Mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)* (pp. 186–195).

Zilis, S., & Kenny, D. (2016). *Why Data Is The New Oil*. Fortune's annual Brainstorm Tech conference. Retrieved from http://fortune.com/2016/07/11/data-oil-brainstorm-tech/