Classification and Prioritization of Tasks in Public Administration

Raúl Isea [1,*]

[1]Fundación Instituto de Estudios Avanzados IDEA, Hoyo de la Puerta, Baruta, Venezuela

Abstract

A large volume of data is being generated in public administration and it is necessary to develop new computational methodologies to classify and analyze it to do a better analysis and decision making. For this reason, the goal of this paper is to present a computational methodology that allows classifying and prioritizing a series of complaints using Artificial Intelligence techniques. To test this model, we generate 600 complaints in four sectors of the public administration to prove the concept. Later, we calculated the tree decision with the help of the Confusion Matrix, and finally the Priority Matrix (based on the Eisenhower model) allows setting priorities on the complaints, and offers the possibility of delegating and even postponing the response to them. In this way, it is possible to prioritize the complaints made in the public administration.

Introduction

We continually make decisions and many of them are based in unconscious emotions. Moreover, there are large volumes of information because it is hard to identify the root of the problem and a lot of them come from a common problem. A hypothetical example is the following: we considered, for example, the case in which all mortgage applications received by a bank and these must be analyzed very carefully. The criteria of these analyses depend on the bank's policy, such as the loan will only be possible as long as the applicant has seniority equal to or greater than three years, with a higher monthly income to a level defined by the institution, and the person must not have a criminal record.

The bank's decision (or not) to grant credit would be relatively fast while the total number of credit applications are few, but if the number is very high, the process of granting credits is very slow. For this reason, we need to implement a computational algorithm that helps to make all decisions assertively in the shortest possible time.

The same process should be occurred in the public administration, but the advances in information technology can help us prioritize over other requests that help improve public services. Remember that the decisions must obey the legal norms according to the priorities set by a certain country because it is a political decision to prioritize the response time of public complaints received. For example, a political decision may be to address problems regarding the water supply in a community instead of public transportation.

For this reason, the present work develops computational methodologies that allow automatic decision making, thanks to advances in algorithms developed in Artificial Intelligence (9). Among these methodologies is an algorithm called decision tree, which can be visualized as a flowchart that implies a series of logical decisions that allow us to solve the problem.

When conducting a search in the scientific literature, we should highlight the work of Bosin (1) the need to prioritize activities in public administration because that not all activities should have the same priority. Schinkel et al (8) develops a methodology based on decision making according to principal–agent framework and thus keep track of activities. Gadson (4) indicated the need to talk to the community for decision making in the COVID-19 Crisis. Recently, Madani and Ghorbanizadeh (2023) they prioritized the public administration problems in the Islamic Republic of Iran based on interviews with experts, key points, and literature. In these as well as other publications, they do not develop a computational methodology for decision making.

The next section describes the computational methodology to record the different activities that must be prioritized in public administration.

Qualitative description of the program

This paper employs the scheme based on the decision tree. A decision tree begins with a root node (called a decision node), and later the different intermediate nodes are automatically generated as the attributes are evaluated (thanks to the calculation of entropy) according to the degree of randomness of the data (9). Thus, the best solution is the one with the lowest value of entropy. The node with the lowest entropy value is called the terminal.

However, decision trees lead to errors by small changes in the original data, an effect known as the overfitting phenomenon (2). To avoid this, an algorithm called pruning (ref) must be implemented, which allows avoiding new branches or eliminating some of them (this last algorithm will be implemented in future works).

The next step is to evaluate the prediction performance of the decision tree based on supervised learning, and for this we calculated the Confusion Matrix (2). Remember that this matrix analyze the data and grouped into four variables called True Positive, True Negative, False Positive, and False Negative (2), ie., that True Positive refers to when the tree predicts to whom the loan is granted and indeed the bank grants the loan.

A True Negative is when the bank denies credit and the program predicts it will not be extended. False Negative is when the bank granted the loan, but the tree predicted that it would not, and False Positive occurs when the loan is not granted, but the system indicated that it would. In the example discussed above, the False Positives and False Negatives are zero; while the True Positives were 224 and the True Negatives were 76 (figure 1).

It remains to indicate the calculation of the Prioritization Matrix (also called criteria matrix), which allows us to order the complaints depending on whether the degree of Importance and Urgency has been catalogued. In this sense, the definition based on the Eisenhower Matrix (6) is used, which, as we know, was designed to distinguish tasks that must be resolved in the very short term and also allows them to be postponed or delegated. For example, it is preferable to urgently solve a problem of electrical system failure instead of paving a road that gives access to a house.
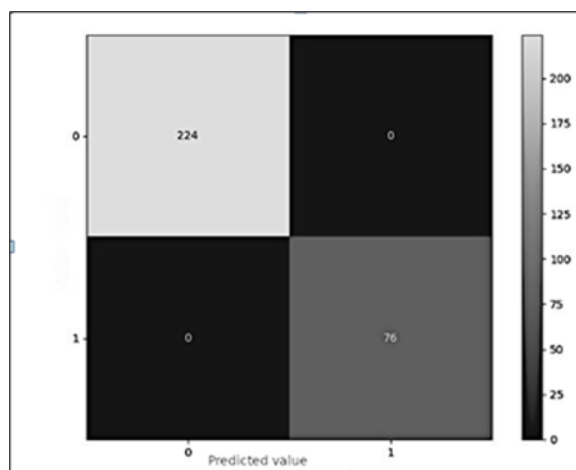
Figure 1. Confusion Matrix obtained from the credit evaluation according to the results

The assignment of priorities will be of two types: Important and 'Not Important'. Furthermore, what is Urgent, but 'Not Important', should be delegated to other institutions for its rapid solution; while those complaints that are not important (and also 'Not Urgent') can be eliminated from the list of complaints to give priority to those that are.

In the next section, some technical details of the computational methodology proposed in the work to classify and prioritize decision-making where 600 complaints were randomly generated in four different sectors are described, but the technical details will be sent to another journal specializing in Artificial Intelligence algorithms.

Computational methodology

The computational implementation was developed in Python, and divided into four parts. First, the proposed computational methodology is indicated to prioritize decision-making through the use of the Confusion and Priority matrices, and then it is indicated how the records of the complaints were generated.

Decision tree calculation

The decision trees are based on a non-parametric supervised learning algorithm, that is, the data learns iteratively, without fixing the number of variables necessary for decision making (7) according to Hunt's algorithm, which consists of the division into subsets where the optimal inflection points in the tree are sought (3).

The degree of entropy (or the Gini index) is usually calculated at each node, which indicates the percentage of incorrect classification when an attribute is randomly selected and, therefore, the lower the value, the closer the correct answer will be.

Confusion matrix

Once the prediction of the tree is made, it is necessary to determine its performance, and for this the Confusion Matrix is calculated, which allows us to determine the accuracy, precision and sensitivity of the decisions obtained with this methodology (3). In fact, it is possible to calculate the accuracy of the prediction and also the precision and sensitivity as described in the book by Mitchell (6).

Prioritization matrix

Prioritizing decisions is made through the Prioritization Matrix in order to adjust the results, according to the political guidelines set by a public administration. The advantage of defining this type of matrix is that those complaints or tasks are managed according to the budget and available resources, allowing the importance and degree of urgency to be quantitatively calculated, where the first is a subjective measure that indicates how that decision affects in the action taken, while the second is the time required for the prompt resolution of the complaint.

Generating and grouping complaints

The results of a survey with 600 completely randomly generated records were simulated, whose data were distributed in different sectors where the complainant's data is stored. First, starts with the person's data (i.e., name, address, telephone numbers, age, sex, marital status). Subsequently, all the complaints were categorized corresponding to public service problems, and we have only considered problems about: drinking water supply, electrical service, roads, public transport, gas distribution, and telephone service, the others are placed as Others. Finally, Urgent, Non-Urgent, Important, and Not Important labels are added according to public service problems. In the case of a specific complaint, it was indicated as 'Not Urgent' and 'Not Important', so the system automatically did not consider it.

Results

The simulation of the complaints has been done randomly with the help of a random number generator where all complaints have the same weight without favoring any complaint. Those that were focused on drinking water problems, electrical service cuts, and road and public transportation problems were classified as Urgent; those of another nature were considered non-urgent (this decision is a political decision and fix in the program).

Later, the program reads the file with the 600 complaints and it calculated the decision tree and later the Priority Matrix (figure 1) which determined the order in which these complaints should be made. The calculation was carried out in the Python programming language with the help of Sklearn libraries (the details will be published in a future work).
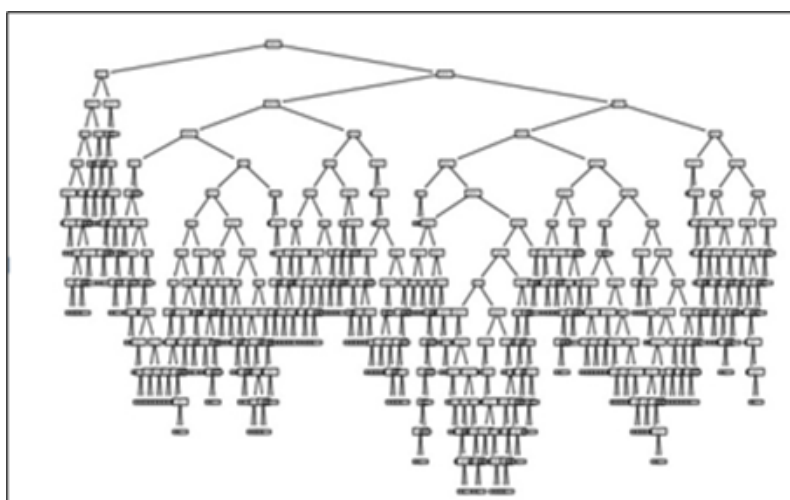

Figure 2. Decision tree obtained after analyzing 600 complaints in the public administration

The figure 2 shows the classification of the complaints in the four municipalities, cataloged by public service, where the represents problems related to drinking water, while 2, 3 and 4 correspond to complaints derived from the electrical system. The results shown in figure 2 have been identified in bars for a quick visualization, and the blank spaces correspond to the complaints that were discarded as they were 'Not Important'.
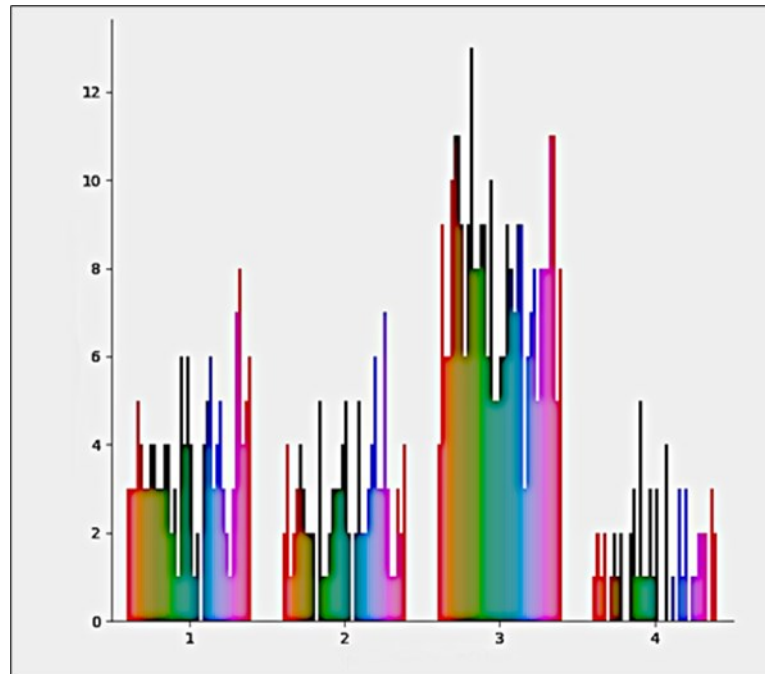

Figure 3. Classification of complaints in the four municipalities, listed by public service; it means, complaints by (1) drink water, (2) electrical systems, (3) roads and public transport, and (4) others.

The result of the evaluation of the Priority Matrix is identified in figure 3, where the first bars indicate those complaints to which a quick response must be given because they are Urgent and Important, while those that must be delegated to other institutions are at the end of it. To avoid confusion, the system will give a list of complaints in order from highest to lowest priority, but they were omitted in the work, these are totally random and only useful for the development of the program.

Conclusions

The computational methodology proposed in this work allows prioritizing decision-making on complaints made, although it can also be extended to tasks or activities. In fact, the degree of precision of the results obtained from the tree is determined with the help of the Confusion Matrix, while the Priority Matrix (based on the Eisenhower model) allows setting priorities on the complaints, and offers the possibility of delegating and even postpones the answer to them.

Acknowledgment

References

1. Bosin, M. R. (1992). Priority setting in government: Beyond the magic bullet. Evaluation and Program Planning, 15(1): 33-43.

2. Breiman, L., Friedman, J. H., Olshen, R. y Stone, C. (1984). Classification and regression tree. Pacif grove: Wadswordth & Brooks.

3. Chikalov, I. (2011). Average time complexity of decision trees. Berlin Heidelberg: Springer-Verlag.

4. Gadson, D. N. (2002) Advancing Equity in Public Administration: Prioritizing Equality of Outcomes in the COVID-19 Crisis. Risk Hazards Crisis Public Policy. 11(4): 449–457.

5. Madani, J., & ghorbanizadeh, V. (2023). Identification and prioritization public administration problems in the Islamic Republic of Iran. Public Management Researches (available at https://jmr.usb.ac.ir/article_7324.html)

6. Mitchell, T. (1997). Decision Tree Learning. En T. Mitchell, Machine Learning (pp. 52-78). USA: The McGraw-Hill Companies, Inc.

7. Murthy, S. K. (1998). Automatic construction of decision trees from data: a multi-disciplinary survey. I Data Mining and Knowledge Discovery, 2(4): 345-89.

8. Schinkel, M.P., Tóth L, Tuinstra, J. (2020) Discretionary Authority and Prioritizing in Government Agencies, Journal of Public Administration Research and Theory, 30(2): 240–256.

9. Smith, C. (2017). Decision trees and random forest: a visual introduction for beginners. Blue Windmill Media.