ROC APP: AN APPLICATION TO UNDERSTAND ROC CURVES

George VON BORRIES[1]
Allan Vieira de Castro QUADROS[2]

■ ABSTRACT: We present a software application (`https://gfvonborries.shinyapps.io/roc_app/`) to help students understand the Receiver Operating Characteristic (ROC) curve and other concepts associated with binary classification models. We use the diagnostic test scenario as a motivation to explain the underlying concepts and the app functionalities. The ROC App enables students to interactively learn why/how the ROC curve closely relates to the accuracy rates, by seeing how these curves and rates respond to modifications on the population's parameters.

■ KEYWORDS: Teaching ROC curve, sensitivity, specificity, accuracy.

1 Introduction

The ongoing Coronavirus-19 (COVID-19) pandemic has brought to surface several statistical concepts that used to be disregarded by the general audience. Diagnostic testing terms such as true and false positive rates suddenly became frequent in every newspaper and conversation, rivalling with more daily topics such as sports and politics. The ordinary citizen has understood that different tests will likely present accuracy discrepancies depending on the underlying methodology. For the COVID-19, for example, it is known that the antigen tests are faster but less accurate than polymerase chain reaction (PCR) ones (WATSON *et al.* 2020).

Despite the increase in the popular knowledge on these topics, we shall not overlook the fact that, even among statisticians, it is not rare to encounter misinterpretations of the underlying theory of diagnostic tests and its supporting concepts, especially including the Receiver Operating Characteristic (ROC) curve.

[1]Universidade de Brasília - UnB, Departamento de Estatística, CEP: 70910-900, Brasília, DF, Brazil. Email: *gborries@unb.br*
[2]Kansas State University, Department of Statistics, 66506, Manhattan, KS, U.S.A. Email: *quadros@k-state.edu*

These misinterpretations clearly have their origins in the undergraduate level. From our experience in teaching categorical data analysis in statistics and health undergraduate programs, it is noticeable that some students usually struggle to understand how exactly the ROC curve works and how it relates to the model accuracy.

To overcome situations like the aforementioned ones, instructors have been recurrently resorting to the use of web applications as an auxiliary tool for teaching statistics (FAWCETT, 2018; ARNHOLT, 2018; FREIRE, 2018). In this article, we present a web based software application (app) conceived as an educational tool to help instructors and support students in the learning process of the statistical theory associated with ROC curves and the accuracy of binary classification models. The app is valuable to instructors and students. Instructors have an easy and attractive way to introduce the theory and students have an intuitive and interactive tool to help them grasp more intricate concepts. Students can modify the underlying population parameters in order to see how the ROC curve shape and the discriminating power of the classifier are affected by each re-parameterization. We observed this to be a highly effective tool to support the learning processes in ROC curves.

2 The ROC app

In this section, we present the general visual aspect of the app along with its main components.

The ROC App could be acessed at `https://gfvonborries.shinyapps.io/roc_app/`. It consists of a dashboard-like panel (Figure 1) developed under the Shiny framework (CHANG *et al.*, 2020) using the R programming language (R CORE TEAM, 2020). Users will find two main sections: the control panel populated with four slider buttons on the left side; and, on the righ-hand side a large plotting area with three plotting panels. Basically, almost every interaction with the slider buttons will cause a reaction on the plotting panels. Users can also download the original R code from `https://github.com/GvBorries/ROCApp`.

The *Distributions Panel* on the top-left of the plotting section illustrates the population distributions (successes and failures) and the chosen discriminating point between them. The successes distribution is plotted in green whereas the failures are plotted in blue. Students can increase or decrease the separation of these distributions by moving the green and/or the blue sliders. These are the controls responsible for mean values of the distributions. The orange slider controls the theoretical-optimal discriminating point between the given distributions.

On the top righ-handed of the *ROC Curve Panel*, we have the resulting Receiver Operating Characteristic curve. This curve helps the user to find optimal separation threshold, which is controlled by the purple slider button on the control panel. The shape of the curve will only be affected by the separation magnitude of the population distributions. Although we will elaborate more on this later, once we adjust the optimal discriminating point on the *Distributions panel*, the ultimate goal is to match the purple dashed crossing line to the orange one on the ROC curve.

Finally, the *Confusion Matrix Panel* on the bottom presents an interactive and hybrid crosstable/graphic object, showing the model outcomes for the given distributions parameterization. Any increase or decrease in the distributions separation or any modification in the discriminating point will affect the predictive power of the binary classification model, hence reflecting on the outcomes presented on this panel.

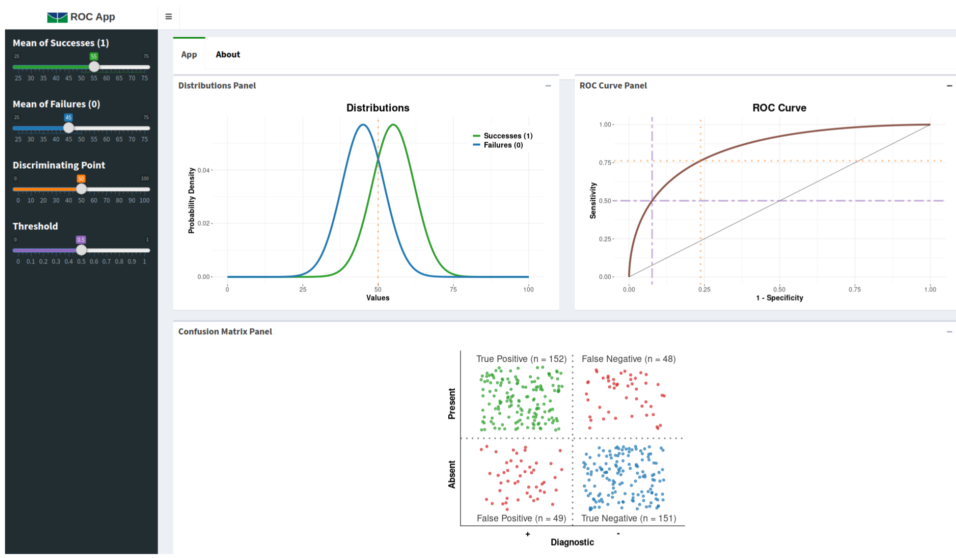

Figure 1 - Full view of the ROC App graphical user interface. The plotting area (right-hand side) reacts to modifications on the slider buttons on the control panel (left-had side).

3 Underlying theory and concepts

The present app tries to fill a gap in the teaching and learning process of the theory related to ROC curves and model accuracy. Hence, we believe that some commentary on these and other related concepts are crucial to explain how the app works. In this section, we briefly present a walk-through on the basics of binary classification problems along with illustrative examples of how the ROC App could support the understanding of the associated theory.

3.1 True and False Diagnostics

Diagnostic tests, such as the ones for COVID-19, can usually be reduced to a binary classification problem in which we have two populations of interest: people that are infected by the virus, and those that are not. In Statistics, the presence of an event of interest is usually called a *"success"* whereas the absence is called a *"failure."*

The primary goal of the underlying statistical model associated with the diagnostic test is to discriminate between these two populations with the maximum accuracy possible. Moreover, researchers want the test to perform better on the infected than on non-infected population. This is so because it is more worrisome to indicate a negative outcome when the tested patient actually has the disease than indicate a positive outcome when the tested patient does not have the disease. The aim is to reduce the probability of false negative outcomes, thus preventing cases in which infected people believe and act as they do not have the disease, hence increasing the virus spread.

The possible outcomes of a binary classification problem are a false-negative (FN), a false-positive (FP), a true-negative (TN) or a true-positive (TP). A false negative case occurs when the test result is negative, but the patient actually has the disease. A false-positive case, on the other hand, consists of a wrong indication of the disease when the patient is not infected. The true-positive and true-negative cases are the successful outcomes of the test. When the underlying model correctly predicts a positive case, we have a true-positive outcome. Similarly, when the test correctly indicates the absence of the virus, we have a true-negative case.

The discriminating power of the diagnostic test (and hence of the statistical model) is evaluated considering the rates of the successful outcomes. Here, two other terms commonly used by statisticians and epidemiologists come into consideration: specificity and sensitivity. The specificity, also known as the true-negative rate (TNR) consists of the proportion given by the number of true-negative cases discriminated by the model over the total size of the healthy population. The sensitivity, by its turn, also called the true-positive rate (TPR), consists of the proportion of the true-positive outcomes with respect to the size of the infected population. These relations can be described by the formulas

$$specificity = \frac{TN}{TN + FP},$$

and

$$sensitivity = \frac{TP}{TP + FN}.$$

Ideally, the two populations (infected/success and healthy/failure) would be sufficiently well separated so that a discriminating point for the test value could be chosen in a way that the model would be capable of perfectly predicting any outcome. This is the hypothetical case in which our model would have both sensitivity and specificity equal to 1, resulting in 100% accuracy.

However, the real-world is a bit more complicated, which means that both distributions with respect to the test value are not sufficiently separated. On the contrary, they have overlapping regions where, despite the chosen discriminating value, we will always have false-positive and false-negative outcomes (WATSON *et al.*, 2020). Thus, any statistical test will present a trade-off between specificity and sensitivity. If a modification on the discriminating value causes the model to better predict the infected population (successes), we will have a consequent increase in sensitivity (TPR) in exchange to a reduction in the specificity (TNR), and vice-versa.

Figure 2 illustrates a change on the discriminating point, moved further to the left-hand side on the discriminating point slider button, which increases the true-positive outcomes and decreases the true-negative ones in comparison to the default parameterization presented in Figure 1. The model would theoretically classify any observation with test value greater or equal to the discriminating point as pertaining to the successes population (green). Note that moving the discriminating point to the left causes the model to accurately predict a greater proportion of the successes distribution in exchange to a smaller proportion of the failures, as shown on the *Confusion Matrix Panel* in Figure 2. This illustrates the trade-off between the sensitivity and specificity.

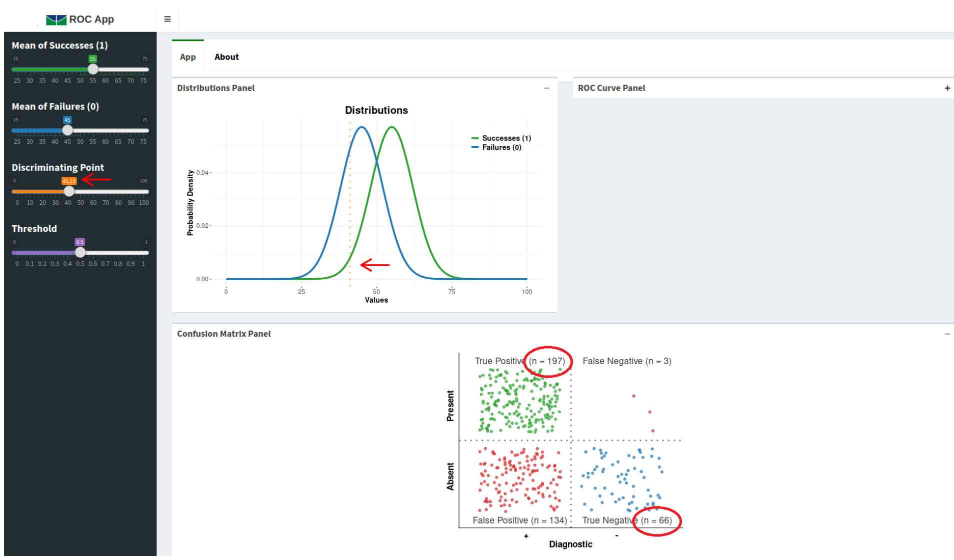

Figure 2 - Trade-off between sensitivity (TPR) and specificity (TNR) as a consequence of shifting the discriminating point to left-hand side. The *ROC Curve Panel* is omitted as we intend to focus only on the modifications of the plots in the first and third panels.

3.2 ROC curve and related concepts

Defining a discriminating value for a test is not an easy task. As a matter of fact, it cannot be done since no one really knows the real distributions of the test values on both populations. Therefore, statisticians and epidemiologists have to resort to the model outcomes in order to find a value that results in the most effective separation.

The ROC curve can assist the data scientist to find the optimal discriminating point. The ROC curve indicates the trade-off between the proportion of true-positive outcomes (sensitivity) and the false-positive rate (FPR) at different classification thresholds (Nettelman, 1998). Note that this is a slightly different trade-off in comparison to the one mentioned in the last section (TPR vs TNR), and is also a very common source of misinterpretations among students and even professionals. The false-positive rate or FPR is the complementary value of the specificity, i.e. the value given by $1-$specificity, and it is also called fall-out or false-alarm ratio. Therefore, the ROC curve gives us the trade-off between the proportion of true-positive outcomes and the model's false-alarm ratio.

Students are usually taught that, if no further specification is made, the optimal discriminating threshold will be the point of greatest distance to the baseline of the ROC curve. However, this information, simply put in that way, does not help us to clearly understand the practical meaning of this choice, thus impairing the understanding of the classification problem in its entirety. Our ROC App aims to overcome this difficulty by allowing students to interactively find the optimal threshold by themselves.

The app is designed in such a way that students are able to choose the threshold based on the discriminating point that, in theory, would produce the optimal separation between the populations under scrutiny. Moreover, they can play with the population parameters and the discriminating point to interactively visualize the influence of these choices on the the ROC curve. Thus, students are instigated to move the threshold orange slider in order to match the optimal separation of the populations, as illustrated on Figure 3. By doing so, students can easily verify that, under the assumption of true-positive and true-negative outcomes being equally important, the optimal discriminating threshold will always lie in the point of greatest distance to the baseline of the ROC curve (see Figure 3).

However, one shall not forget that diagnostic tests are expected to perform better on the true-negative cases. Thus, we will have different weights for the performance on the true-negative against true-positive cases, consequently affecting the trade-off between these two measures, and between the sensitivity and the fall-out as well. In that case, the optimal threshold will lie slightly off the point of the greatest distance to the baseline.

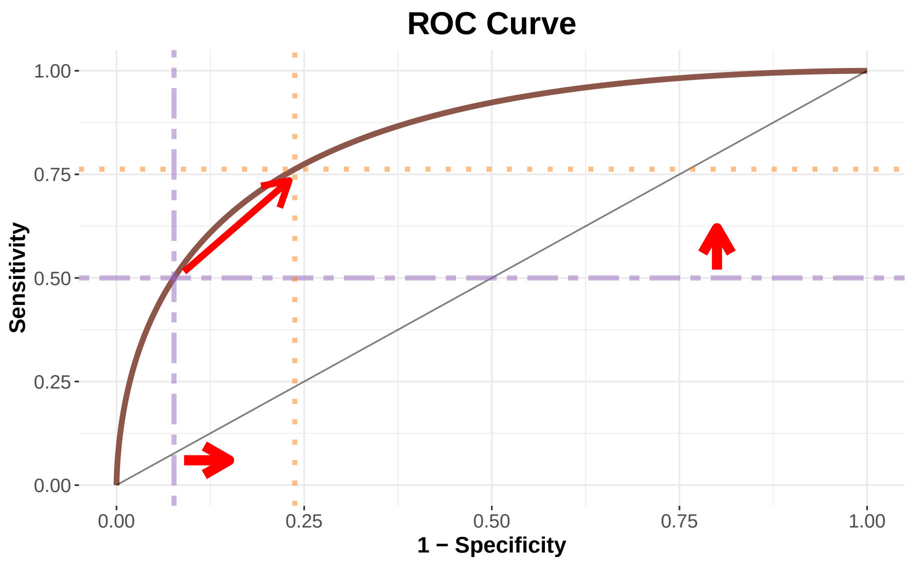

Figure 3 - Receiver Operating Characteristic (ROC) curve and the threshold motion toward where is the optimal discriminating point.

3.3 Shape and area under the curve

The ROC curve can also be deployed to compare different models. For this aim, we use the Area Under the Curve (AUC) measure. Basically, the greater the AUC of a model the higher the accuracy.

The ROC App also facilitates students' understanding of the AUC concept. We observe that the greater the separation between the distributions, the easier is to discriminate between the cases, and the better is the model performance. At the same time, students can observe that an increase in the separation of the distributions results in an increase in AUC, as illustrated in Figure 4.

Figure 4 - Area Under the Curve (AUC). The AUC increases from top to bottom as the separation between the distributions increases. This is attained by moving the population slider buttons apart from each other on the control panel.

Conclusions

The ROC App has been tested inside classes with very auspicious results. It has demonstrated to be a powerful complementary resource to a textbook and explanatory lectures, since students are able to interactively explore the concepts and theory of binary classification problems in further depth. Not only the concepts are learned more easily, but also the underlying relationships among them are straightforwardly grasped by students with the support of the app.

The fact that the user does not need to know the R programming language in order to use the app certainly facilitates its introduction to a broader audience of students with very different backgrounds. However, we recommend instructors to ask students to also explore the source code of the app by encouraging them to introduce modifications on it. Adding a slider button to control the variance of the populations would be a very interesting starting point, for instance. This could be a good motivational reason for students to explore basic concepts related to the R language and the Shiny framework which are not addressed in this paper.

Acknowledgments

The authors are grateful to Dr. R. von Borries from the University of Texas at El Paso, reviewers, and editors for their suggestions that improved greatly the final version of our paper.

VON BORRIES, G.; QUADROS, A. V. C. App ROC: Um aplicativo para entender curvas ROC. *Braz. J. Biom.*. Lavras, v.40, n.2, p.213-222, 2022.

- *RESUMO: Um aplicativo é apresentado (`https://gfvonborries.shinyapps.io/roc_app/`) para ajudar estudantes a entender a Curva Característica de Operação (curva ROC) e conceitos associados com modelos de classificação binária. Utilizamos um cenário de teste diagnóstico como motivação para explicar os conceitos envolvidos e algumas funcionalidades do aplicativo. O App ROC permite que estudantes aprendam de maneira interativa porque e como a curva ROC está relacionada com taxas de acurácia através da visualização de como estas curvas e taxas respondem a modificações nos parâmetros populacionais.*

- *PALAVRAS-CHAVE: Ensino da curva ROC, sensibilidade, especificidade, acurácia.*

References

ARNHOLT, A. Using a shiny app to teach the concept of power. *Teaching Statistics*, v.41, 2018.

CHANG, W.; CHENG, J.; ALLAIRE, J. J.; XIE, Y.; MCPHERSON, J. *Shiny: Web application framework for r.* R package version 1.5.0. 2020.

FAWCETT, L. Using interactive shiny applications to facilitate research-informed learning and teaching. *Journal of Statistics Education*. v.26, n.1, p.2-16, 2018.

FREIRE, S. Using shiny to illustrate the probability density function concept: Probability density function. *Teaching Statistics*, v.41, 2018.

NETTLEMAN, M. D. Receiver operator characteristic (roc) curves. *Infection Control and Hospital Epidemiology*, v.9, n.8, p.374-377, 1988.

R CORE TEAM. *A R: language and environment for statistical computing*. Vienna, Austria. 2020. `https://www.R-project.org/`.

WATSON, J. W.; WHITING, P. F.; BRUSH, J. E. Interpreting a covid-19 test result. *British Medical Journal*, v.369, n.1808, 2020.

Received on 08.07.2021.

Approved after revised on 15.10.2021.