v. 1, n. 1, 2024

International Journal of
Exact
Sciences

LEARNING OBJECTS FOR STUDYING THE MATHEMATICAL FRAMEWORK OF MACHINE LEARNING CLASSIFICATION TECHNIQUES AND THEIR PERFORMANCE EVALUATION METRICS

Rubens Dias Rodrigues Junior
Mackenzie
https://lattes.cnpq.br/7131737626934959

Ivan Carlos Alcântara de Oliveira
Mackenzie
http://lattes.cnpq.br/0272400268231478

Abstract: A learning object (LO) is any educational resource intended to support students' study and learning. They are reusable and can incorporate text, graphics, animations, audio, and video. Based on the above, this work carried out the design and development of three LOs to guide those interested in the subject to understand the mathematical components of the KNN (K-Nearest Neighbors), Naïve Bayes, and SVM (Support Vector Machines) algorithms using Python as a coding platform and Jupyter Notebook pages containing text, graphics, and videos to enrich the learning experience.
Keywords: Learning Objects, Machine Learning, SVM, KNN, Naive Bayes

INTRODUCTION

Learning objects are educational resources that can be used to support the teaching and learning process in different educational contexts, such as schools, universities, corporate training and online courses. According to Wiley (2006), learning objects are "any digital resource that can be reused to support teaching". In this sense, there is a wide variety of resources in different formats that can be worked on, for example, in the form of video, animation, podcast, simulation or a combination of these. These resources can be used to present complex concepts in a visual and interactive way, providing an engaging learning experience.

Using learning objects as a resource, the problem of mathematical understanding in the study of machine learning algorithms (ML) is presented. The search for information on how machine learning models and their metrics work does not always return very didactic and easy-to-understand information, which raises the opportunity to create an educational resource to assist in the mathematical framework and the application of these techniques in the most diverse business scenarios currently required. However, the complex literature, technical materials and courses available lack effective teaching methods, containing mathematical language that is difficult to understand (MASOLA, 2019, as cited in SELBACH, 2010, p. 40) that often do not explore other means of presenting knowledge, such as visual resources and simulations.

In view of the above, this article intends to explore the construction of digital educational resources using different formats to provide mathematical knowledge and the applicability of supervised ML models and their metrics to assist in the studies of data science professionals and students interested in this topic.

The algorithms selected for the study were K-NN (K-nearest neighbors), Naive Bayes and SVM (Support Vector Machines). The repository of the objects selected was the Jupyter notebook and the Python programming language.

The content of this article is presented in the following sections. Section 2 presents the theoretical framework that addresses the concepts of learning objects, supervised ML techniques with illustrations of the mathematical framework and performance metrics, as well as some ML algorithm simulators. The methodology used, containing a summary of the steps, is found in Section 3. Section 4 presents the results obtained, highlighting the three LOs produced and some final considerations. Finally, Section 5 highlights the final considerations and proposals for future work.

THEORETICAL FRAMEWORK

INTRODUCTION

In his thesis, Souza (2020) states that Mathematics teaching focuses on the product, on ready-made formulas, and states that: "in most educational institutions, especially in relation to Mathematics teaching, the greater emphasis is placed on the product to the detriment of the process, resulting in poor quality of the former" (SOUZA, 2020, p. 23).

This finding is no different when we enter the field of data science. Business problems are becoming increasingly complex, just as the volume of data has been increasing, which drives the development of more sophisticated computational tools based on Machine Learning (FACELI, 2021, p.2), which involve more complex algorithms that are difficult to understand mathematically. Taking this issue into account, as well as a greater demand for data science professionals in the market, there are a significant number of training courses in this area being offered, aiming to teach the application of artificial intelligence in business scenarios, but without focusing on the process of applying the algorithms themselves or the mathematical process involved, but only offering an execution pipeline, which generates the emergence of data science professionals who execute algorithms, nevertheless without adequate mathematical and procedural knowledge of what they are doing.

When faced with real problems in a company or research, questions such as "Which algorithm to use?," "Which evaluation metric(s) to use?," "How to understand the metrics?", "Where to start?," become very common and searching for literature is a challenge. One of the consequences is the significant use of digital forums, which have a "garb" of digital educational resources, which have become one of the main learning resources for developers of computational algorithms today.

LEARNING OBJECTS

How to define a Learning Object? Although there are many definitions, the following stand out:

> "Any entity, digital or non-digital, that can be used, reused, or referenced during the use of technologies that support teaching." (WILEY, 2006)

In order to identify the characteristics that a learning object must have, Santórum et al. (2021) conducted a literature review and found that there is, in fact, no consensual definition among the authors surveyed regarding learning objects. This study established that a learning object is a digital, independent, reusable, interoperable resource with an educational purpose. In addition, it must be found with simple search algorithms and be evaluated for its effectiveness, cost, and usability. This approach was adopted in this research project.

Digital teaching resources have been a motivating element for students in their teaching-learning process, whether due to their innovative appeal or the interaction that these resources provide. Learning objects arise from the need to apply these resources, taking advantage of their potential for reuse, portability, classification and identification.

One of the first difficulties when working with learning objects is the multiplicity of resources available for their design. Designs have varied from the specification of the nature of the object (digital/non-digital), its size (minimum, extensive), its characteristics (three-component structure/free structure), its open/closed condition, among others (CASTRO; DURÁN, 2008).

Since these are digital resources that complement learning, individual skills and the different ways of learning of students must be considered when designing a learning object. This is one of the potentials of using learning objects, as they facilitate multiple

International Journal of Exact Sciences ISSN 2966-1153 DOI https://doi.org/10.22533/at.ed.153112401074

means of representing and transmitting pedagogical content: texts, images, audios, videos, animations, and games (OLIVEIRA, 2018). It also promotes the personalization of teaching resources according to the student's needs and learning style.

Considering accessibility, learning objects can support assistive technologies (devices or resources that help people with disabilities to conducted their activities independently), effectively expanding access to teaching content.

In the areas of Mathematics and Data Science, the demand for learning objects seems to be driving the development of digital tools that have been changing the way we learn. In fact, with the advent of the coronavirus pandemic that began in 2019 and the increase in online teaching methods, self-teaching is increasingly necessary to keep up with studies. However, the use of technology must be valued for the nature of the knowledge that can be constructed and not just because it can provide motivation or facilitate learning (TRIANA-MUÑOZ et al, 2016). In Mathematics, for example, several resources are found on the internet to support students, of which I would like to highlight:

- Geogebra (Geogebra, 2023): Tool for learning mathematics that offers the possibility of simulations in geometry, algebra, statistical graphs, calculus, etc.
- Symbolab (Sy, 2023): Tool for solving algebraic, trigonometric equations, calculus, etc.
- Derivative Calculator (Derivative Calculator, 2023): Tool for calculating differential equations step-by-step.
- Integral Calculator (Integral Calculator, 2023): Tool for calculating integrals step-by-step. As examples of the usefulness of this type of resource, some simulation sites were found on the internet, also called playgrounds, where it is possible to learn machine learning models in a didactic way through simulation, adding value to learning, such as:
- Tensorflow Playground (Maruseac, 2023): It is possible to simulate the parameters of a neural network to understand how it works.
- Time-Series Playground (Encora, 2023): Time series simulator that makes it possible to test parameters, decompose series, evaluate metrics, and obtain mathematical knowledge involved in the algorithms.

SUPERVISED MACHINE LEARNING MODELS

According to Faceli (2021), ML tasks can be divided into Predictive and Descriptive. Predictive models can be used, for example, to predict a patient's health status based on their symptoms. In these tasks, ML algorithms that follow the supervised learning paradigm are generally used (FACELI, 2021, p. 3).

Within the machine learning category, we have classification models. Such models consider known labels (also called tags or categories) in their training stage with the aim of learning patterns to make better predictions on future data (test data or production data).

According to Olson (2017), the following algorithms deserve attention:

- Naive Bayes;
- Logistic Regression;
- Decision Tree;
- SVM: Support Vector Machine;
- Random Forest: Random Forests;
- K-NN: Nearest Neighbors;
- XGBoosting.

In this work, the K-NN, Naive Bayes and SVM algorithms were selected. This choice took into consideration, the varied mathematical characteristics involved in solving these algorithms. KNN involves calculating the distance between points (analytical geometry), Naive Bayes is based on probability and SVM uses elements of linear algebra.

Mahesh (2020) provides a description of these algorithms, namely:

- **K-nearest neighbors:** The K-NN algorithm is a simple, supervised machine learning algorithm that can be used to solve classification and regression problems. It is easy to implement and understand, but has the major disadvantage of becoming significantly slower as the size of the data being used increases;

- **Naive Bayes:** It is a classification technique based on Bayes' Theorem with an assumption of independence between predictors;

- **Support Vector Machine (SVM):** SVMs can efficiently perform non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

KNN (K-NEAREST NEIGHBORS)

The K-Nearest Neighbors (KNN) model is a machine learning algorithm used for classification and regression. It is one of the simplest and most intuitive methods for classifying new data points based on their proximity to existing examples.

KNN works quite simply. It stores a training data set with known labels and when it receives a new unlabeled example, it searches for the k nearest neighbors in the training set. The distance between the points is calculated using a metric such as Euclidean distance and the k nearest neighbors are selected.

In the case of classification, the most common label among the k neighbors is assigned to the new example. In the case of regression, the average of the labels of the k neighbors is used as the predicted value for the new example.

The value of k is an important parameter in KNN. A smaller value of k makes the model more sensitive to noise and fluctuations in the data, while a larger value smooths the decision boundaries and may lose fine details.

KNN has some limitations such as the need to keep the entire training set in memory to perform nearest neighbor queries, which can be computationally expensive for large datasets. In addition, choosing the right distance metric and k value is also important to obtain accurate results.

In summary, KNN is a simple and effective machine learning algorithm that classifies new examples based on their proximity to training examples, making it suitable for classification and regression problems.

According to Faceli (2021), KNN has the following advantages and disadvantages:

Advantages:

- Simple training algorithm;

- Simplification of objective function for optimization;

- Applicable to simple and complex problems;

- Incremental algorithm for training purposes.

Disadvantages:

- Greater computational effort in the training phase;

- To classify a new observation, it is necessary to calculate the distance of this object in relation to all training objects.

Mathematically, the KNN algorithm can be explained by applying the Pythagorean theorem through a simple definition of analytical geometry, as complemented by Faceli in his literature through Equation 1.

$$|PQ| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad \text{(Equation 1)}$$

To make the mathematics more convenient, we can generalize the formulation and describe the Euclidean distance, or distance between two points, as illustrated in Equation 2.

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{d} \left(x_i^l - x_j^l\right)^2} \qquad \text{(Equation 2)}$$

Figure 1 shows an illustrative example of the application of KNN with K=1, dividing the class into two classes.

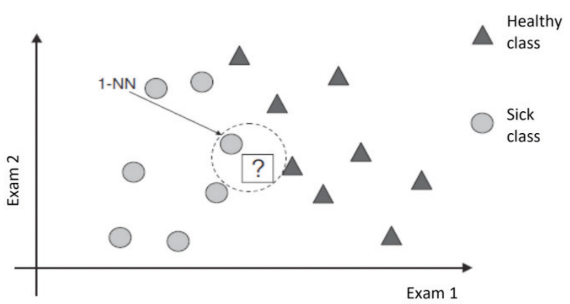

Figure 1: Example of 1-NN. K=1

Source: Faceli (2021)

NAIVE BAYES

The Naive Bayes algorithm is a probabilistic classification method based on Bayes' theorem. It is widely used in text classification tasks such as sentiment analysis, spam detection, and document categorization. The Naive Bayes algorithm assumes that features are independent of each other, which may not be true in practice. However, even with this simplification, Naive Bayes can be highly effective in many cases. There are variations of the Naive Bayes algorithm, including Multinomial Naive Bayes, Bernoulli Naive Bayes, and Gaussian Naive Bayes. Each variation is suitable for diverse types of data and classification problems. According to its mathematical formulation, the Naive Bayes algorithm starts from the premise of a priori probability to find a posteriori probability of an observation belonging to a certain class within the available observations, as explained in Equation 3 (Mahesh, 2020).

$$P(c|x) = \frac{P(X|C)P(c)}{P(x)} \qquad \text{(Equation 3)}$$

When applying Bayes' formula, we can find extremely low probability values. To prevent this behavior, also called underflow, the formulation is increased by the log function, as explained by Faceli (2021) in Equation 4.

$$\log(P(y_i \mid \mathbf{x})) \propto \log(P(y_i)) + \sum_j \log(P(x^j \mid y_i))$$

(Equation 4)

According to Jadhav et al (2016), the advantages of the Bayes algorithm are its performance, it requires little computational resources in training, and it improves classification performance by removing irrelevant variables. However, for the model's results to be good, the Naive Bayes classifier requires a large database and also presents less accuracy for some types of datasets.

SVM (SUPPORT VECTOR MACHINES)

Support Vector Machine (SVM) is a supervised machine learning algorithm that is often used for classification and regression. It works by finding a hyperplane that separates data into different classes or that best fits the data in a regression.

Basically, the SVM algorithm creates decision margins in a hyperplane, and these margins are defined by what we call a support vector. These vectors will determine the decision boundaries for an observation to belong to one class or another, as we can see in Figures 2 and 3, found in (Faceli, 2021).

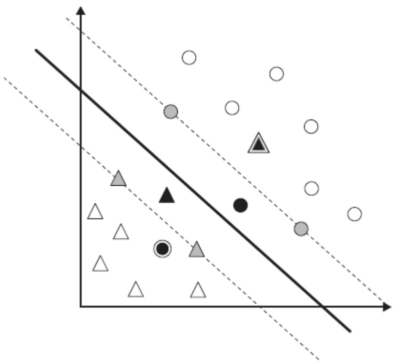

Figure 2: Linear SVM

Source: Faceli (2021)

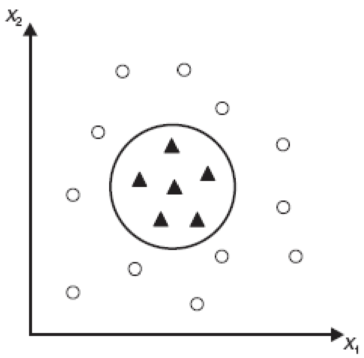

Figure 3: Non-Linear SVM

Source: Faceli (2021)

Simplifying the mathematics in this algorithm, we have the formulation Equation 5 and Equation 6 to understand the SVM algorithm (Winston, 2010).

$$(w, x_n) + b \geq 0 \; when \; y_n = +1 \qquad \text{(Equation.5)}$$

$$(w, x_n) + b < 0 \; when \; y_n = -1 \qquad \text{(Equation 6)}$$

Translating Equation 5 and 6, the SVM algorithm calculates the scalar product between vectors and depending on the direction of the vector in relation to the support vector determined by the algorithm, it can classify an observation into its appropriate class.

SVM is a popular machine learning algorithm due to its ability to manage complex and high-dimensional data. However, its sensitivity to hyperparameters and its difficulty in dealing with large data sets are some of the disadvantages that need to be considered when choosing SVM as a classification method (Zhang, 2021).

PERFORMANCE METRICS

The task of evaluating classification models is to measure the degree to which the classification suggested using the model corresponds to the actual classification of the case (NOVAKOVI´C et al, 2017). Depending on the observation method, there are different measures for evaluating model performance. The selection of the most appropriate measures must be made depending on the characteristics of the problem and the ways in which it is implemented. Among the metrics that can be evaluated, some stand out, considering that it is not possible to establish the best techniques a priori, since it depends on the problem to be studied (FACELI, 2021, p. 148). The following metrics, for example, are calculated from a matrix called "confusion matrix," which, according to Düntsch (2019), has the format shown in Table 1.

		True value	
		P	***N***
Predicted Value	***P***	***True Positive***	***False Positive***
	N	***False Negative***	***True Negative***

Table 1: Confusion matrix:

Source: Adapted from Düntsch (2019)

From the confusion matrix, according to Castro (2016), the metrics accuracy, Precision, recall and F1-Score can be obtained and are easily understood mathematically.

Accuracy (Equation 7) measures the number of correct classifications divided by the total number of classifications.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad \text{(Equation 7)}$$

Precision (Equation 8) measures the accuracy of the algorithm, that is, the probability that a retrieved item is relevant.

$$Precision = \frac{TP}{TP+FP}$$ (Equation 8)

Recall (Equation 9) measures the completeness of the algorithm, that is, the probability of retrieving an item being relevant.

$$Recall = \frac{TP}{TP+FN}$$ (Equation 9)

F1-Score (Equation 10) is used to evaluate classification performance in a range [0,1].

$$F1Score = \frac{2.Precision.Recall}{Precision+Recall}$$ (Equation 10)

Classification problems still refer to more complex mathematical formulations, bringing more opportunities for creating learning objects to aid teaching, such as a metric called Log Loss, which is a classification metric based on probability (BARRON, 2019).

MACHINE LEARNING PLAYGROUNDS

It is now common to find Machine Learning algorithm simulators available on the internet, and these have been called Machine Learning Playgrounds. These environments are intended to simulate algorithm behaviors according to fictitious data created by the user at the time of execution. Some interesting examples are: Encora Time Series Playground for simulating time series, Tensor Flow Playground for simulating neural networks, and other simpler examples such as Machine Learning Playground (ml-playground), which simulates the most common algorithms we know, such as KNN, SVM, decision tree, among others.

These more abstract support applications have become very important as learning support, as well as for more agnostic testing of ML models. One example is CFU playground, an open-source framework for machine learning acceleration for developing hardware acceleration for neural network processing. This full-stack framework gives users access to explore experimental and custom architectures that are customized and optimized for embedded ML (PRAKASH et al., 2023). According to the definitions of our study, these simulators could fit into the concept of digital Learning Objects.

METHODOLOGY

This research was applied in nature and involved the development of artifacts in the form of learning objects, or digital educational resources, for the study of the mathematical framework of the ML algorithms KNN, Naive Bayes and SVM. The following set of activities were conducted during the period of validity of this project:

- Initial bibliographic research: refers to the research of works in databases of articles, journals, dissertations and theses available on the internet, of private and public content, using as keywords the main concepts considered to be the focus and essential, namely: mathematical learning objects, types of digital learning objects, learning objects, the current state of open educational resources, supervised machine learning techniques, roc curve, evaluation of classification models, matthews correlation coefficient, logarithmic loss machine learning, cosine similarity measure text classification and accuracy precision and recall and f1.

- Reading of researched bibliography: involves reading the bibliography selected in the research carried out in the previous stage;

- Preparation of a conceptual map identifying the main concepts related to the focus of this research;

- Selection of machine learning techniques and their evaluation metrics for development;

- Development of machine learning algorithms for evaluating error and performance metrics;
- Implementation of mathematical formulations of the metrics studied;
- Study and selection of the content and formats of the digital resources that make up the learning object;
- Construction and provision of digital learning objects.

First, a bibliographical research was conducted on digital learning objects and machine learning algorithms and their metrics.

In the development stage of the project regarding the mathematical part, machine learning algorithms were created using appropriate computer programming languages (e.g.: Python, R, etc.) with the objective of studying their error and performance metrics.

Based on the study of the algorithms and their metrics, the mathematical formulation of the studied metrics was made, aiming at a more inclusive didactic approach in the study of machine learning algorithms.

Having a more tangible vision of the complexity of the mathematical problems to be shared, the most appropriate digital resources for the learning objects were defined for a didactic approach to the algorithms studied.

Finally, digital learning objects were created to make the results available, aiming at access by an audience seeking mathematical knowledge of the selected ML algorithms.

RESULTS

In this section, a summary of the results obtained is presented. In Section 4.1, two conceptual maps developed in the study phase of this research are shown. The construction process and the structure of the LOs developed are highlighted in Section 4.2. In Section 4.3, the LO related to KNN is described. The LO on Naive Bayes is presented in Section 4.4. In Section 4.5, the LO of SVM is highlighted. Finally, in Section 4.6, some discussions are carried out.

CONCEITUAL MAP

In the study phase of this research, two conceptual maps were created regarding Learning Objects and Machine Learning, illustrated in Figures 4 and 5. These maps were fundamental to support the necessary knowledge on the theme of this work.

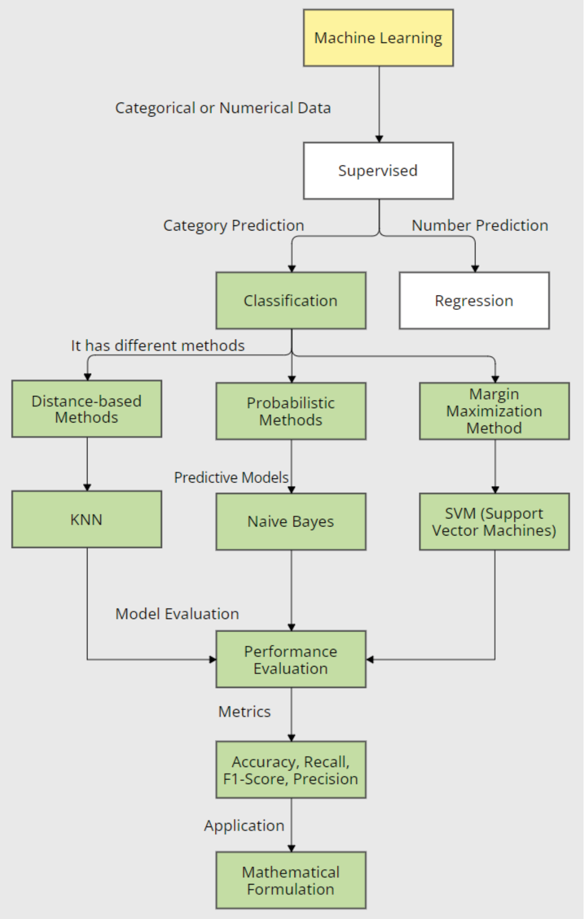

Figure 5: ML Concept Map

Source: Prepared by the author (2023)

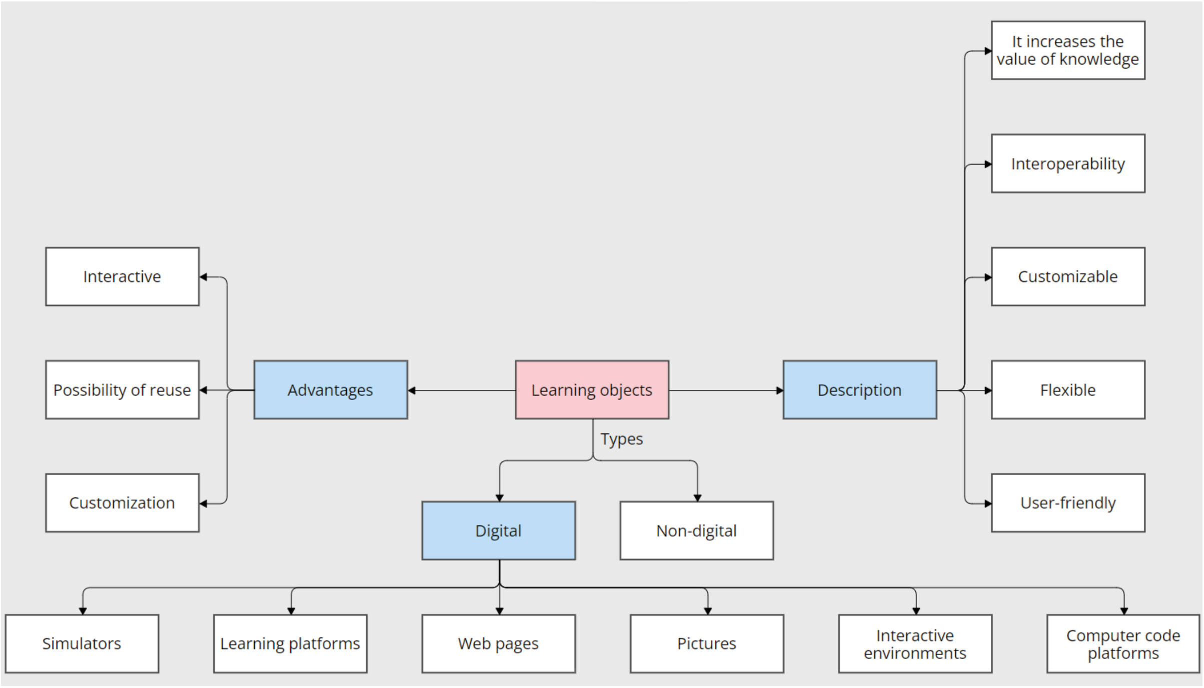

Figure 4: Conceptual Map of Learning Objects

Source: Prepared by the author (2023)

LEARNING OBJECTS: STRUCTURE AND CONSTRUCTION PROCESS

To develop the educational resources for this project, we used Jupyter notebook, a platform used to develop solutions in Python and other programming languages that allows you to view codes and execution results, in addition to incorporating HTML codes, which accepts the addition of videos and images to explain the algorithms. The Python version used was 3.8.0, running in the Anaconda environment manager. The Iris database (Scikit-learn, 2023) was used as the testing database.

Digital educational resources were created for each model studied: KNN, Naive Bayes, and SVM. The structure of code sections and resource formats used followed the same standard sequence for all learning objects, as described below:

- Introduction: Text containing an explanation of the model to be studied;
- Introductory video: Presentation of the learning object and summary of what will be studied;
- Loading the database;
- Exploratory Data Analysis: Use of graphs such as Box plot, parallel coordinates, pair plot, violin, and similar;
- Explanatory video on how to read each type of graph;
- Preparing data to run the ML model;
- Explanatory video on the mathematical framework of the model to be created:
  - KNN Model: Analytical geometry approach, Pythagorean theorem, Euclidean distance and use of a KNN simulator through the ML Playground platform.
  - Naive Bayes Model: Approach to the general formula of the Naive Bayes model with and without the use of logarithm and illustrations.

- SVM (Support Vector Machine) model: Illustration of the concept of margins, scalar product, vector direction and simulation of the model through the Interactive demo of Support Vector Machines platform *(SVM)*.
- Construction of the ML model;
- Evaluation of performance metrics;
- Video explaining the model results and performance metrics;
- Supplementary material: Embedded YouTube videos with mathematical approaches to the models studied in the Learning Objects. One of the most explored channels was StatQuest.
- References.

The developed learning objects are available at https://github.com/rrodriguesrr/learning_objects.git . On the page of this link, pay attention to the README.txt file that contains the minimum instructions for use.

K-NN (K-NEAREST NEIGHBORS):

Following the standard Jupyter notebook section model, the KNN model was created, using the Iris database (Scikit-learn, 2023) as the database. Exploratory data analyses were performed using introductory videos and texts. The model used was the KNeighbors Classifier from the sklearn package. The accuracy metric was used to evaluate the model. Illustrations of the obtained object are highlighted in Figures 6 and 7.

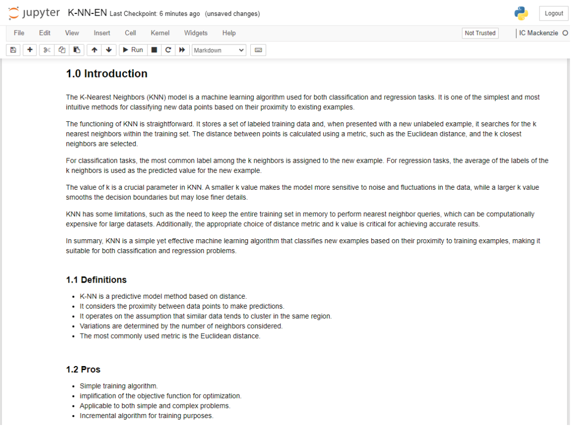

Figure 6: Text-type Learning Object

Source: Prepared by the author (2023)

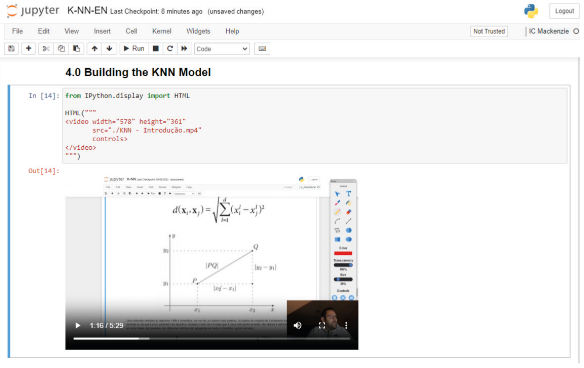

Figure 7: Video-type Learning Object

Source: Prepared by the author (2023)

NAIVE BAYES

The digital educational resource created for Naive Bayes followed the standard Jupyter notebook section model, using Iris (Scikit-learn, 2023) as the database. Exploratory data analyses were also performed, with introductory videos and texts. The model used was Gaussian from the sklearn package. The accuracy metric was used to evaluate the model. Some screenshots of the results obtained for the learning object can be seen in Figures 8 and 9.

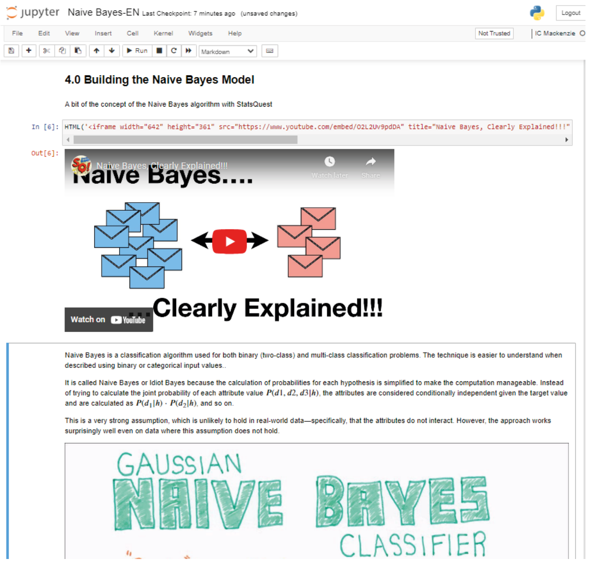

Figure 8: Youtube Embedded Video Learning Object

Source: Prepared by the author (2023)

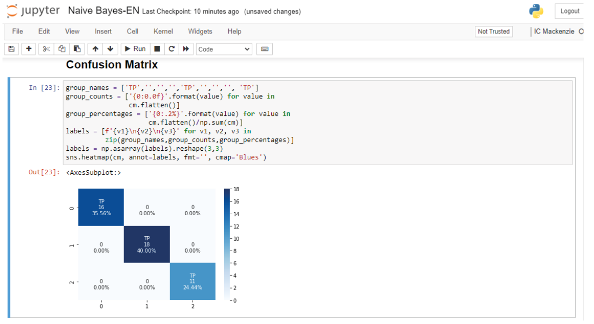

Figure 9: Image/graph type Learning Object.

Source: Prepared by the author (2023)

SVM (*SUPPORT VECTOR MACHINE*)

The construction of the learning object corresponding to the SVM followed the standard sequence described above and the standard Jupyter notebook section model, using Iris (Scikit-learn, 2023) as the database. Thus, exploratory data analyses were performed in this resource, with introductory videos and texts.

The model used was Linear SVC from the sklearn package. The metric used to evaluate the model was accuracy. Examples of object interfaces can be seen in Figures 10 and 11.

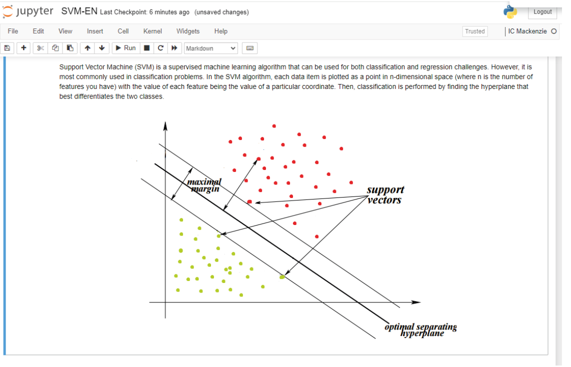

Figure 10: Text and Image Learning Object

Source: Prepared by the author (2023)

Figure 11: Image/Graphic Type Learning Object

Source: Prepared by the author (2023)

DISCUSSION

The use of digital learning objects containing resources with different formats in the exploration of the KNN, Naive Bayes and SVM algorithms and the related mathematical framework provided mechanisms to approach the concepts in a visual and textual manner incorporated into a Jupyter Notebook page with dynamic execution of the models.

The use of videos, images and contextualized textual explanations has the potential to bring differentials, generally not present in books or literature, and has the capacity to assist in the understanding of the studied content, especially in the explanation of the mathematical formulation present in these algorithms.

Since the learning objects were created as a Jupyter Notebook page, Python and its packages, these LOs provide a Storytelling of their execution.

CONCLUSION AND FUTURE PROJECTS

In this work, three learning objects were developed considering the supervised ML algorithms KNN, Naive Bayes and SVM, with emphasis on the mathematical framework that guides their operation and evaluation.

In their construction, a standard sequence of the model generation process was adopted using the Iris database, which, in summary, involved: an introduction to the algorithm, loading the database, exploratory data analysis with graphs and an explanatory video, data preparation and an explanatory video of the mathematical framework, construction of the model and evaluation of the model and an explanatory video of the metric used.

In the LOs, various media formats were also used, including a combination of text, image and video, which were incorporated into a Jupyter Notebook page in Python, with the sklearn package for generating the models. This way, the student can follow the sequence suggested in the LO and interact with it. The interaction occurs through the step-by-step execution of the codes contained in the Python page and the results obtained.

The digital educational resources created are a didactic alternative for the study of ML algorithms, since their design was not focused solely on computing, but also on mathematics teaching methodologies.

Thus, they have the potential to assist people interested in studying and learning this topic and professionals entering the field of data science who are interested in acquiring support to make a more solid argument about the results obtained in their models and the mathematical basis used. However, an evaluation and validation of these LOs with users is necessary to prove their real effectiveness.

This project was presented at WTT 2023 (Technological Trends Workshop) for students and teachers of the Faculty of Computing and Informatics (FCI) of ``*Universidade Presbiteriana Mackenzie*`` (UPM). Through this, it was possible to identify improvements that were added to the resources produced.

Based on the results obtained, the next step would be to select people to conduct an evaluation of their content as a potential facilitator of study and learning. In addition, to produce new LOs with other ML algorithms and other performance evaluation metrics.

THANKS

I would like to thank ``*Universidade Presbiteriana Mackenzie*`` for the opportunity to conduct my first scientific research.

To Professor Dr. Ivan Carlos Alcântara de Oliveira who guided me on this journey not only with his knowledge, but also with his understanding due to my professional situation, which made my time extremely limited to conducted this research.

To Professor Daniela Cunha from the Faculty of Computing and Information Technology who guided me at the beginning of the process and directed me to the supervising professor.

To Professor Eliza Hidemi Sadaike who guided me in my communication with the Faculty of Computing and Information Technology so that it was possible to conduct this achievement.

REFERENCES

BARRON, Jonathan T. **A general and adaptive robust loss function**. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. p. 4331-4339.

CASTRO, Eduardo Peñalosa; DURÁN, Patricia Landa. **Objetos de aprendizaje: una propuesta de conceptualización, taxonomía y metodología**. Revista electrónica de psicologia Aplicada, v. 11, n. 3, 2008.

CASTRO, Leandro Nunes de; FERRARI, Daniel Gomes**. Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações**. São Paulo: Saraiva, v. 5, 2016.

DERIVATIVE CALCULATOR. **Calculate Derivatives online**, c2023. Home page. Available at: < https://www.derivative-calculator.net/ >. Accessed on: sep. 1, 2023.

DÜNTSCH, Ivo; GEDIGA, Günther. **Confusion matrices and rough set data analysis**. In: Journal of Physics: Conference Series. IOP Publishing, 2019. p. 012055.

ENCORA. **Time-Series Playground**, c2023. Home page. Available at: < https://encoradigital.github.io/SAB-timeseries-playground/ >. Accessed on: sep. 1, 2023.

FACELI, Katti; LORENA, Ana C.; GAMA, João; AL, et**. Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina.** 2 ed. Rio de Janeiro: Grupo GEN, 2021.

GEOGEBRA. **GeoGebra for Teaching and Learning Math**, c2023. Home page. Available at: < https://www.geogebra.org/>. Accessed on: sep. 1, 2023.

Greitemann Jonas. **Interactive demo of Support Vector Machines (SVM),** c2023. Home page. Available at: < https://greitemann.dev/svm-demo/>. Accessed on: sep. 1, 2023.

INTEGRAL CALCULATOR. **Calculate Integrals online**, c2023. Home page. Available at: < https://www.integral-calculator.com/ >. Accessed on: sep. 1, 2023.

JADHAV, Sayali D.; CHANNE, H. P. **Comparative study of K-NN, naive Bayes and decision tree classification techniques**. International Journal of Science and Research (IJSR), v. 5, n. 1, p. 1842-1845, 2016.

MAHESH, Batta. **Machine learning algorithms-a review.** International Journal of Science and Research (IJSR).[Internet], v. 9, p. 381-386, 2020.

MARUSEAC, Mihai. **Tensorflow Playground**, c2022. Home page. Available at: < https://playground.tensorflow.org/ >. Accessed on: sep. 1, 2023.

MASOLA, Wilson; ALLEVATO, Norma. **Dificuldades de aprendizagem matemática: algumas reflexões**. Educação Matemática Debate, v. 3, n. 7, p. 52-67, 2019.

ML Playground. **Machine Learning Playground**. Home page. Available at: < https://ml-playground.com/>. Accessed on: sep. 1, 2023.

NOVAKOVIĆ, Jasmina Dj et al. Evaluation of classification models in machine learning. **Theory and Applications of Mathematics & Computer Science**, v. 7, n. 1, p. 39–46-39–46, 2017.

OLIVEIRA, Ivan Carlos Alcântara de. **Recursos Educacionais Abertos e Objetos de Aprendizagem. 2018.** [Supporting PDF file for the postgraduate course at Universidade Cruzeiro do Sul].

OLSON, Randal S. et al. **PMLB: a large benchmark suite for machine learning evaluation and comparison**. BioData mining, v. 10, n. 1, p. 1-13, 2017.

PRAKASH, **Shvetank et al. Cfu playground: Full-stack open-source framework for tiny machine learning (tinyml) acceleration on fpgas**. In: **2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)**. IEEE, 2023. p. 157-167.

Scikit-learn. **The iris Dataset**, c2023. Available at: < https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html/>. Accessed on: sep. 1, 2023.

SOUZA, José Ocimar Barros de. **Modelagem matemática como processo para o desenvolvimento do pensamento analítico e reflexivo**. 2020. Doctoral Thesis.

SANTÓRUM, M., CARRIÓN-TORO, M., GUACAPIÑA, M-B, ACOSTA-VARGAS, P. and JADÁN-GUERRERO, J. "**A Case Study: Developing reusable Learning Objects**," *2021 Second International Conference on Information Systems and Software Technologies* ***(ICI2ST)***, 2021, pp. 79-86, doi: 10.1109/ICI2ST51859.2021.00019.

STATQUEST. **An epic journey through statistics and machine learning**, c2023. Home page. Available at: < https://statquest.org/>. Accessed on: sep. 1, 2023.

SY. **Symbolab**, c2023. Home Page. Available at: < https://pt.symbolab.com/ >. Accessed on: sep. 1, 2023.

TRIANA-MUÑOZ, Mónica María; CEBALLOS-LONDOÑO, Juan Fernando; VILLA-OCHOA, Jhony Alexander. **Una dimensión didáctica y conceptual de un instrumento para la Valoración de Objetos Virtuales de Aprendizaje**. El caso de las fracciones. Entramado, v. 12, n. 2, p. 166-186, 2016.

WILEY, David. **The current state of open educational resources**. 2006.

Winston, Patrick H., **Lecture 16: Learning: Support Vector Machines: Artificial Intelligence MIT OCW**. Fall 2010. Lecture Notes.

Zhang, Y., & Yang, Y. **A Comparative Study of Machine Learning Algorithms for Predicting the Risk of Heart Disease**. Journal of Healthcare Engineering, 2021.