CARCARAQSAR: A FULL-STACK COMPUTATIONAL WEB APPLICATION FOR QUANTITATIVE STRUCTURE–ACTIVITY RELATIONSHIP ANALYSIS

CARCARAQSAR: UMA APLICAÇÃO WEB COMPUTACIONAL FULL-STACK PARA ANÁLISE DE RELAÇÕES QUANTITATIVAS ESTRUTURA–ATIVIDADE

Daniel Alencar Penha Carvalho
ORCID 0009-0009-2637-5998

Colegiado de Engenharia da Computação, Universidade Federal do Vale do São Francisco (UNIVASF)
Juazeiro, Brazil
danielalencar746@gmail.com

Rosalvo Ferreira de Oliveira Neto
ORCID 0000-0002-3290-5539

Colegiado de Engenharia da Computação;
Grupo de Pesquisas em Algoritmos Aplicados à Química Medicinal e Inteligencia Artificial (ALQUIMIA);
Universidade Federal do Vale do São Francisco (UNIVASF)
Juazeiro, Brazil
rosalvo.oliveira@univasf.edu.br

Edilson Beserra de Alencar Filho
ORCID 0000-0002-1000-0114

Colegiado de Ciências Farmacêuticas (CFARM);
Programa de Pós-Graduação em Ciências da Saúde e Biológicas (PPGCSB);
Programa de Pós-Graduação em Biociências (PPGB);
Grupo de Pesquisas em Algoritmos Aplicados à Química Medicinal e Inteligencia Artificial (ALQUIMIA);
Universidade Federal do Vale do São Francisco (UNIVASF)
Petrolina, Brazil
edilson.beserra@univasf.edu.br

Abstract. CarcaraQSAR is an open-source, full-stack web application designed to simplify the development of Quantitative Structure-Activity Relationship (QSAR) models. By integrating machine learning algorithms and bio-inspired feature selection techniques, the tool enables researchers to efficiently identify chemical descriptors correlated with biological activity. Its user-friendly interface eliminates the need for extensive programming knowledge, making QSAR modeling more accessible. CarcaraQSAR supports various validation strategies, including cross-validation and Y-randomization, ensuring robust and reproducible models. Designed for scalability, it allows cloud deployment, overcoming limitations of traditional QSAR tools. This application significantly enhances predictive modeling capabilities, thereby contributing to cost-effective drug discovery and molecular research. The synthesis of important tools in a single flow also makes CarcaraQSAR a teaching tool for University courses in Pharmaceutical and Medicinal Chemistry, providing students with a practical and integrated understanding of the concepts involved in QSAR.

Keywords: computational chemistry; QSAR modeling; machine learning; bio-inspired algorithms; chemistry education.

Resumo. O CarcaraQSAR é uma aplicação web full-stack de código aberto, desenvolvida para simplificar a criação de modelos de Relações Quantitativas Estrutura-Atividade (QSAR). Integrando algoritmos de aprendizado de máquina e técnicas bioinspiradas de seleção de características, a ferramenta permite aos pesquisadores identificar eficientemente descritores químicos correlacionados com atividade biológica. Sua interface amigável elimina a necessidade de amplo conhecimento em programação, tornando a modelagem QSAR mais acessível. O CarcaraQSAR suporta diversas estratégias de validação, incluindo validação cruzada e Y-randomização, garantindo modelos robustos e reproduzíveis. Projetado para escalabilidade, permite implantação em nuvem, superando limitações das ferramentas QSAR tradicionais. Essa aplicação melhora significativamente as capacidades de modelagem preditiva, contribuindo assim para a descoberta econômica de fármacos e pesquisa

molecular. A síntese de ferramentas importantes em um só fluxo torna ainda o CarcaraQSAR uma ferramenta didática para cursos Universitários de Química Farmacêutica e Medicinal, proporcionando aos estudantes uma compreensão prática e integrada dos conceitos envolvidos no QSAR.

Palavras-chave: química computacional; modelagem QSAR; aprendizado de máquina; algoritmos bioinspirados; educação em química.

1. INTRODUCTION

The development of Quantitative Structure−Activity Relationship (QSAR) models has long been a cornerstone of Computational Chemistry and pharmaceutical research (Cherkasov et al., 2014). As the pharmaceutical industry increasingly embraces computational methods to accelerate drug discovery, the demand for accessible, efficient QSAR modeling tools has become paramount. However, the limited availability of comprehensive and user-friendly software solutions is a significant obstacle, particularly for researchers with minimal programming expertise or restricted access to advanced computational tools. While some free software options have been available, they often lack cutting-edge AI integration, usability features, and alignment with established standards such as the OECD Principles for QSAR Models (OECD, 2004). Additionally, most tools require complex setups involving multi-version Python installations and dependencies like Docker, which can create compatibility issues, hindering their adoption in academic and public research environments.

Pharmaceutical and Medicinal Chemistry encompasses the design, synthesis, and development of new therapeutic agents, focusing on understanding molecular interactions and biological activities at the chemical level. Computational chemistry has become an indispensable tool in this field, significantly accelerating drug discovery by predicting molecular properties, biological activities, and potential therapeutic effects through advanced modeling techniques (Supuran, 2019). The integration of computational platforms and specialized software into Pharmaceutical and Medicinal Chemistry education is particularly crucial, as it bridges theoretical knowledge with practical applications, enhancing students' ability to visualize complex biochemical processes, design effective molecular structures, and engage actively in drug discovery research (Supuran, 2019). Such integrated educational tools prepare students to proficiently address real-world challenges in drug development and pharmacological innovation.

Several QSAR modeling tools have been developed to support drug discovery and molecular design, with software applications available in both free and paid formats. Among the free platforms, QSARINS (Gramatica et al., 2014) and Qsar.co (Ambure et al., 2019) are noteworthy examples, offering essential features for constructing QSAR models. On the other hand, widely used commercial tools like Schrodinger (Schrodinger, 2023) and BIOVIA Discovery Studio (Dassault Systèmes BIOVIA, 2016) provide more comprehensive solutions but are associated with high costs. A common limitation also found is their exclusive reliance on local installation, which necessitates significant computational resources and technical configuration, restricting access for users with limited infrastructure or expertise.

QSARINS (Gramatica et al., 2014) is a software designed for the development, analysis, and validation of QSAR models. It emphasizes rigorous external validation and strict adherence to OECD principles, key features for ensuring reliable predictions of chemical activity. Specifically, QSARINS exclusively employs multiple linear regression (MLR) with Ordinary Least Squares (OLS), which can restrict its applicability to more complex modeling scenarios. Additionally, the software is only compatible with desktop environments, which may limit its flexibility and accessibility for modern, cloud-based workflows. It is freely available for academic use, but these constraints should be considered when evaluating its suitability for diverse research needs.

QSAR-Co (Ambure et al., 2019) is open-source software for developing multi-target classification-based QSAR models, adhering to OECD guidelines and employing Linear Discriminant Analysis (LDA) or Random Forest (RF) for robust predictions. It includes tools for data validation, applicability domain analysis, and model prediction, with a user-friendly workflow. Its functionality is limited to one nonlinear modeling technique (RF), a single variable selection algorithm (Genetic Algorithm, GA), and implementation as a standalone desktop application, which may restrict its flexibility and scalability for advanced research needs.

Another associated resource, the QSAR-Co-X platform, is an open-source computational toolkit also designed to support the development of multi-target QSAR models by integrating classical QSAR principles with modern machine-learning strategies (Halder and Cordeiro, 2021). The software enables the simultaneous modeling of multiple biological targets within a unified framework, allowing the identification of compounds with polypharmacological profiles. QSAR-Co-X incorporates descriptor calculation, feature selection, model validation, and applicability domain analysis, following international best practices for QSAR modeling. By facilitating multi-target analysis, the toolkit addresses key challenges in drug discovery, such as data sparsity and target selectivity, and provides a flexible and transparent environment for reproducible QSAR studies.

AlvaScience, in turn, is a software suite developed to provide complete support for the QSAR workflow, with specific application in blood-brain barrier (BBB) permeability. The platform integrates descriptor calculation, model development, statistical validation, and applicability domain analysis, offering a unified and reproducible environment for predictive studies of relevant pharmacokinetic properties (Mauri and Bertola, 2022).

NanoBRIDGES is an open-access toolkit for the development of QSAR and nano-QSAR, enabling the modeling of biological and toxicological properties of nanomaterials. The software is designed to handle conventional and nanospecific descriptors, facilitating predictive analyses in nanotechnology and risk assessment, with an emphasis on transparency and methodological standardization (Ambure et al., 2015).

OCHEM (Online Chemical Modeling Environment) (Sushko et al., 2011) is an open-access, web-based platform that allows users to build and validate QSAR models without requiring installation or programming expertise. It integrates a database of curated experimental data and a modeling framework that supports various machine learning algorithms, including support vector machines and neural networks. OCHEM is especially valuable for its emphasis on data transparency, automated model validation, and reproducibility, which are essential for regulatory applications and scientific rigor. However, the requirement for internet connectivity and account creation may limit its usability in environments with restricted access.

KNIME (Konstanz Information Miner) (Berthold et al., 2009) is a graphical analytics platform that supports QSAR modeling through a modular, node-based workflow. Its flexibility and integration with cheminformatics libraries such as RDKit make it a powerful tool for both novice and advanced users. KNIME enables users to perform complex data preprocessing, feature selection, and modeling using a visual programming interface, which helps bridge the gap between experimental scientists and data scientists. Nevertheless, its broad general-purpose design requires careful setup and customization for specific QSAR applications, which may pose a learning curve for users seeking out-of-the-box solutions.

The application presented here was called “CarcaraQSAR”, an acronym for “Computational Algorithms to Relate Chemical Attributes with Respective Activities”, also paying homage to a bird of prey typical of the Brazilian caatinga biome – “Carcará”. Unlike traditional desktop-based tools, its cloud-ready platform ensures scalability and minimizes the need for advanced local infrastructure. The platform also provides a comprehensive suite of

artificial intelligence techniques, including Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Linear Regression, ensuring flexibility for diverse modeling scenarios. Additionally, CarcaraQSAR incorporates advanced variable selection algorithms, such as Genetic Algorithm and Artificial Bee Colony, enhancing the precision and reliability of predictive models. This accessibility is complemented by advanced features such as integrated model validation techniques, including Leave-one-out, Cross Validation K-Fold, and Y-randomization, which are not often seen together in existing tools. By combining usability, flexibility, and robust evaluation methods in a single environment, CarcaraQSAR facilitates broader adoption and supports efficient research and development workflows.

In this scenario, CarcaraQSAR emerges as an alternative and user-friendly platform designed to simplify the construction of regression models for QSAR analysis, especially considering laboratories routines that work with small sets of molecules and without familiarity with programming, as well as didactic utility in Medicinal Chemistry, bioinformatics and related classes. It offers a full-stack web application integrating advanced multivariate analysis techniques, real-time performance metrics, and internationally recognized standards. Its simple design empowers chemists and pharmaceutical researchers to perform sophisticated modeling without requiring extensive programming knowledge. By bridging accessibility and high functionality, CarcaraQSAR facilitates the development of predictive models that reduce experimental costs and accelerate drug discovery processes. In the educational context, CarcaraQSAR can be used to facilitate the understanding of abstract concepts and also allows immediate practical application, essential in academic training in disciplines such as Pharmaceutical Chemistry/Medicinal Chemistry. The software may be accessed online through a web-based interface, or alternatively installed and executed on local machines or institutional servers, with support for scalable deployment in cloud computing environments.

2. METHODS

2.1 General presentation

The CarcaraQSAR application is built for portability, featuring Django REST (Python) as its backend and ReactJS (JavaScript) as its frontend, ensuring compatibility with major architectures. The hardware requirements include a mid-range processor (e.g., Intel Core i5), 8 to 16 GB of RAM, and a 256 GB to 1 TB solid-state drive. For cloud deployment, CarcaraQSAR can be easily implemented on platforms like Amazon AWS. Detailed installation guidelines and documentation are available in its GitHub repository (https://GitHub.com/rosalvoneto/CarcaraQSAR). This architecture ensures stable performance even under heavy computational loads, making it reliable for large-scale research projects.

CarcaraQSAR is a robust and versatile application specifically developed to support the construction of Quantitative Structure-Activity Relationship (QSAR) models. The platform features an intuitive interface that streamlines the customization of data preprocessing and model generation workflows, addressing the diverse needs of researchers. By leveraging advanced algorithms, CarcaraQSAR intelligently identifies and selects the most relevant variables from the data set, ensuring that QSAR models are trained using features with the highest predictive significance. During the model training process, the application automatically generates validation metric plots, providing real-time feedback on model quality and predictive performance. This dynamic evaluation process empowers users to refine their models effectively, enhancing their accuracy and overall reliability while contributing to advancements in computational chemistry and pharmaceutical research.

In addition to its technical-scientific use, CarcaraQSAR is specially designed for pedagogical application, allowing students to interact directly with advanced computational

modeling techniques, thus improving teaching and learning in university disciplines related to medicinal and pharmaceutical chemistry.

The following sections provide a comprehensive analysis of CarcaraQSAR's architecture, data-flow, and operating environment. Each aspect is detailed to offer insights into the platform's design, functionality, and practical applications.

2.2 CarcaraQSAR software architecture

The architecture of CarcaraQSAR, as illustrated in Figure 1, is built on a robust integration between a Django-based backend and a React.js frontend. User interactions with the system occur through the React.js interface, which processes user requests efficiently and provides seamless access to the CarcaraQSAR platform. To manage data and credentials, the software utilizes an SQLite3 database. The backend's Django API is served by Gunicorn (Green Unicorn), a high-performance WSGI HTTP server optimized for UNIX environments. Web requests are routed to Gunicorn via a socket managed by the Nginx web server, which acts as a reverse proxy, efficiently handling static files and managing web traffic.

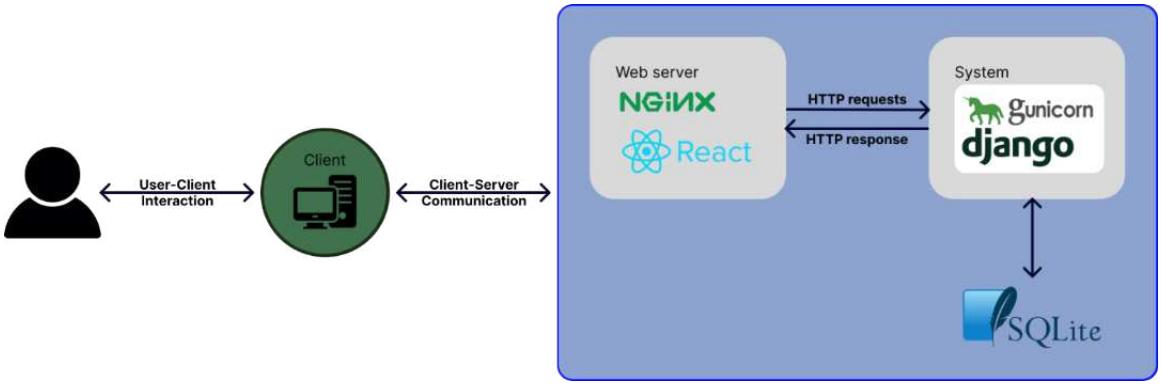

Figure 1. System architecture of the CarcaraQSAR platform. The user accesses the system through a client interface developed with React.js, served by the Nginx web server. HTTP requests are processed by the backend, which is implemented in Django and managed by the Gunicorn WSGI server. Data and results are stored in a SQLite database.

To ensure operational continuity in a production environment, the system employs systemd, a process management tool specific to Ubuntu systems, which guarantees the uninterrupted execution of CarcaraQSAR's essential services. Communication between the client (React.js) and the server (Django API) is facilitated through HTTP requests (POST, GET, PUT, DELETE), where the server processes the requests and returns the appropriate data to the frontend. The frontend then displays this data to the user, ensuring a smooth user experience.

This architecture establishes a solid foundation for CarcaraQSAR's features by enabling efficient data exchange and seamless interaction between system components, thereby ensuring reliable and optimized platform operation.

2.2 Software setup and installation

The following link to GitHub shows the entire installation guide and steps required for the operation of the application: *https://github.com/rosalvoneto/CarcaraQSAR*

2.3 Overview of access and project creation at CarcaraQSAR

CarcaraQSAR is designed to serve a diverse audience within the educational, scientific and industrial communities, particularly professionals involved in drug discovery, chemical

research, and environmental studies. Upon accessing CarcaraQSAR for the first time (http://www.carcaraqsar.com.br/), the user is directed to the login screen, where they can either log in to an existing account or create a new one. The registration process requires the following information: name, email, country, institution, and password. After successfully logging in, the user is redirected to the main screen, also referred to as the home screen. On this screen, users can view all previously created projects and start new ones. Each project in CarcaraQSAR follows a structured workflow that begins with importing a database of molecular descriptors and culminates in the creation of a QSAR model. Each project is uniquely linked to a single database that serves as the foundation for the modeling process.

The final QSAR model can be a regression ou classification, the choice between them depends on the nature of the response variable. Regression is applied when this the is continuous and quantitative, such as pIC_{50}, logP, solubility, or LD_{50} values, allowing the model to predict numerical outcomes based on molecular descriptors (Cherkasov et al., 2014). As your turn, classification methods are used when the response variable is categorical, such as binary activity labels (active/inactive) or toxicity profiles (toxic/non-toxic), for example.

Continuous variables can also be transformed into categorical classes through thresholding, for example, by defining compounds with $pIC_{50} \geq 4$ as active and those with lower values as inactive, facilitating classification-based screening. While this discretization simplifies interpretation and may mitigate experimental noise, it can also lead to information loss. The basic algorithm for regression is the Multiple Linear Regression. Other algorithms as Support Vector Machines, Random Forest, can be used for both regression or classification [1].

The first step in this workflow is project creation. To create a project, the user needs to provide only the project name and description. Once created, the project is displayed on the home screen, where it can be deleted or restored if necessary. After creating a project, the user follows the data processing workflow, as depicted in Fig. 2, which illustrates the flowchart of CarcaraQSAR. The table 1 shows all the functionalities available in platform. Each step in this process is detailed in the subsections below.

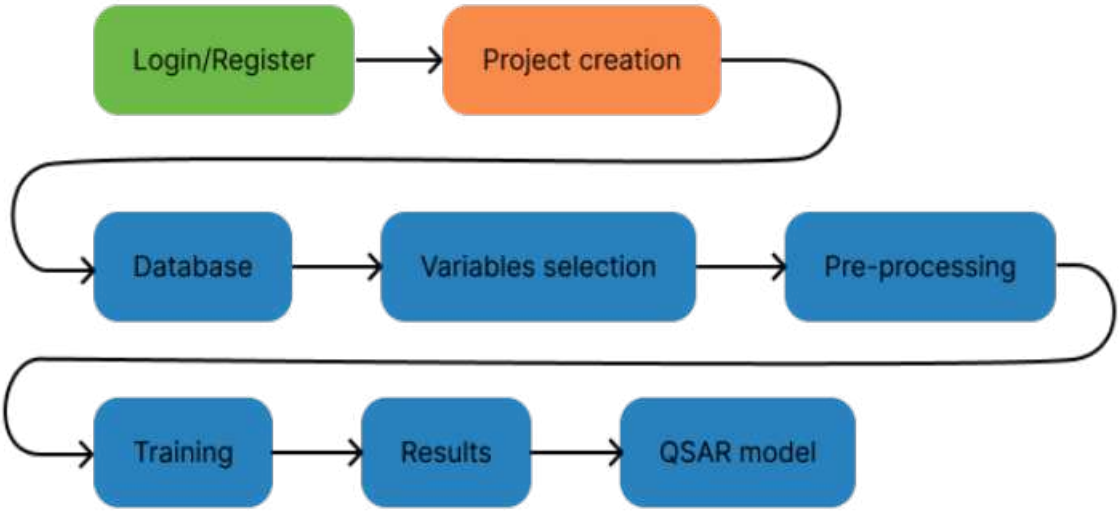

Figure 2. CarcaraQSAR's workflow, from the login and creation of the project to the generation of the QSAR model. Each step represents an essential phase in the development of the model, including variable selection, pre-processing, training and results analysis.

Table 1. Options available during the flow on the CarcaraQSAR application.

Pre-processing	Variables selection 1	Variables selection 2	Base Model
Do not apply	Remove variables (manual)	Do not apply varibles selection algorithm	Random Forest
Min-Max Scaler	Automatically remove constant variables	Genetic Algorithm	Support Vector Machines
Standard Scaler		Bee Colony Algorithm	K-Nearest Neighbors
Robust Scaler			Linear Regression
Normalize			

2.4 Database

In this step, the user must upload a CSV file in which each row represents a molecule and each column corresponds to a 2D descriptor (Bahia et al. 2023). Alternatively, the user can upload a file in SMILES format. If a SMILES file is provided, CarcaraQSAR automatically generates a corresponding CSV file containing molecular descriptors using the PaDEL API (Yap, 2011). Regardless of the uploaded file type, the last column is interpreted as the 'target' variable (Y) for QSAR modeling.

2.5 Pre-processing

In this step, users can explore the dataset variables through visualizations such as histograms and box plots, providing valuable insights into the data distribution and potential outliers. Additionally, normalization techniques can be applied to standardize the variables, ensuring they have equal weight during the modeling stage and improving the overall performance and reliability of the QSAR model.

The available normalization methods are: The Min-Max Scaler, that linearly transforms features to a fixed range, typically [0, 1], preserving the relationships among values but being sensitive to outliers; The Standard Scaler, that standardizes features by removing the mean and scaling to unit variance, producing a distribution with a mean of zero and standard deviation of one (it is suitable when descriptors follow a Gaussian-like distribution); The Robust Scaler, that is designed to reduce the influence of outliers by centering the data using the median and scaling according to the interquartile range, making it appropriate for skewed or noisy data; The Normalize method, that rescales the feature vectors individually to unit norm (L1 or L2), emphasizing the direction rather than the magnitude of vectors, which is particularly useful in distance-based algorithms such as k-nearest neighbors and support vector machines. Selecting an appropriate normalization strategy is essential to improving model performance, convergence, and interpretability in predictive modeling workflows.

2.6 Featuring selection

This step is divided into three main components:

Variable Removal: Users can remove constant variables or other unwanted variables from the dataset. This step ensures the modeling process focuses on meaningful features, improving the quality of the analysis.

Application of Bioinspired Algorithms: Users can choose between genetic algorithms and bee colony algorithms to identify the most relevant variables. Parameters such as the number of iterations, population size, and mutation rates are customizable, enabling fine-tuning to match the dataset's specific characteristics and requirements.

Database History: After completing the variable selection process, users can review the dataset's version history and choose which version they would like to continue working with, whether the original dataset or the reduced variables after selection. This feature provides flexibility, allowing users to revert to previous database versions if necessary.

2.7 Traning

In this step, the user utilizes the selected database to choose a modeling technique, such as Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), or Linear Regression. After selecting the technique, the user specifies the corresponding hyperparameters to optimize the model's performance. Once these configurations are completed, the training process begins, and the user waits for its conclusion.

2.8 Results

In this step, users can apply a variety of validation and analysis techniques to thoroughly evaluate the model's performance. The following performance evaluation metrics and methods are available:

Variable Importance: Calculates and displays a ranking of the most significant variables contributing to the model's predictions, helping to interpret the model's behavior.

Leave-One-Out (LOO): Utilizes a cross-validation technique where each instance in the dataset serves as a single validation example, offering an exhaustive evaluation of the model's performance.

K-Fold Cross-Validation: Divides the dataset into k subsets, training and validating the model iteratively on various combinations of these subsets, ensuring a balanced and robust performance evaluation.

Y-Scrambling: Tests the model's robustness by randomly shuffling the target variable (Y), thereby assessing its capacity to identify real patterns rather than random correlations.

Bootstrap: Creates multiple resampled datasets to train several models, enabling the estimation of the variability and reliability of the predictions, which is crucial for model confidence.

These validation techniques are consistent with the guidelines established by the Organisation for Economic Co-operation and Development (OECD) and ensure adherence to international standards for QSAR model development and evaluation.

2.9 Prediction

After completing the validation process, users can proceed to generate the final QSAR model. At this stage, the following actions are available:

Perform numerical predictions: Based on the selected values of the final variables, predictions can be executed seamlessly through a user-friendly interface. This is facilitated via a form, eliminating the need for any additional configuration or activity from the user.

Download the trained model and associated scaler: Users can export the model and its scaler for future use or integration into external workflows.

Delete and recreate the model: If necessary, users can delete the existing model and restart the process to create a new one.

This step is designed to ensure flexibility, usability, and efficiency, enabling users to refine and apply their models effectively in various predictive tasks.

2.10 System Architecture and Environment Configuration

To ensure the successful operation of CarcaraQSAR, specific software environments and dependencies must be properly configured. The application is developed using Python 3.10 and is structured with the following components:

Backend: Built with the Django 4.2.7 framework, the backend relies on several essential Python libraries, including: a) Django CORS Headers 3.7.0; b) Django REST Framework 3.14.0; c) Django REST Framework Simple JWT 5.3.0; d) Gunicorn 20.1.0; e) Pandas 2.1.4

Frontend: Developed using React.js, the frontend seamlessly integrates with the backend to deliver a comprehensive user experience.

All dependencies required for the backend can be installed via the requirements.txt file, while dependencies for the frontend are managed through the package.json file. Proper installation of these dependencies is critical to ensure the smooth operation of CarcaraQSAR. Failure to correctly configure the environment may result in execution errors or system failures.

3. RESULTS

To demonstrate the functionality of the proposed application, we utilized three different datasets available on our GitHub repository (https://GitHub.com/rosalvoneto/CarcaraQSAR): The firts was one specifically designed to analyze the behavior and effectiveness of Machine Learning algorithms, when addressing small-scale data and regression tasks (Oliveira Neto, 2023); the second corresponding to 55 compounds from our research group (dos Santos et al., 2018), modeled again in an unprecedented way with the CarcaraQSAR tool; the third corresponding to a data set of hundreds of molecules reported in the literature, used here to build a predictive model of solvation free energies, useful in molecular dynamics simulations (Mobley & Guthrie, 2014).

3.1 First example

In this case, we employed the dataset file F25_S50_YL.csv, which contains 50 samples with 25 input variables, of which only 5 (20%) are relevant, and a linear target.

The experiment involved the use of a genetic algorithm for feature selection and linear regression for both variable selection and final model construction. Figure 3 illustrates the relative importance of the variables following feature selection and model training. Feature importance was computed as the mean decrease in impurity across all trees, also known as "Gini Importance."

Model validation was performed using Leave-One-Out (LOO) cross-validation (Figure 4) and K-Fold cross-validation repeated 50 times (Figure 5). The results demonstrate the robustness and predictive accuracy of the model, with a coefficient of determination Q^2_{LOO}=0.808 and K-FOLD values exceeding 0.6. Additionally, the Y-scrambling analysis (Figure 6) confirmed the absence of chance correlations, as all Q^2_{LOO} values for scrambled Y datasets were significantly lower than 0.4.

Finally, Figure 7 displays the user interface, where researchers can predict values for new patterns using the trained model.

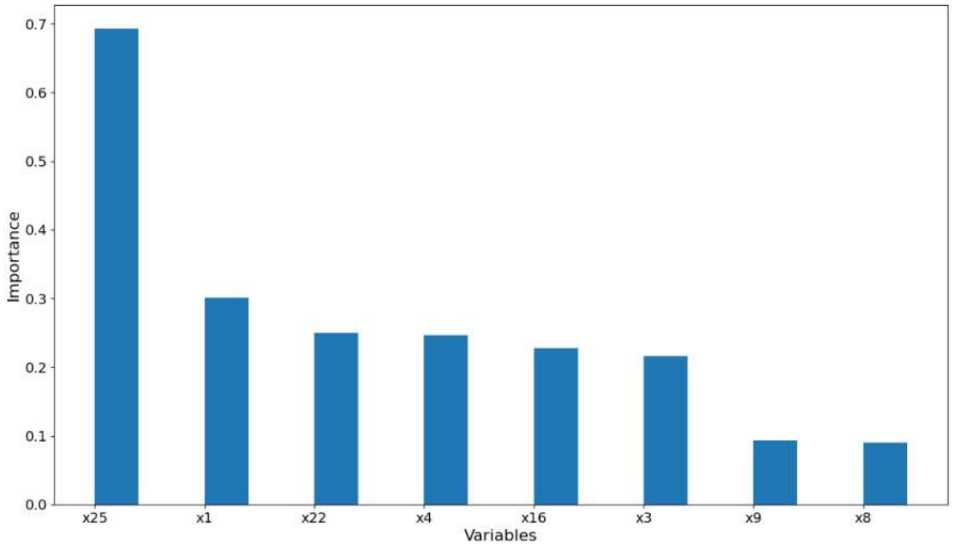

Figure 3. Relative importance of variables after feature selection and model training, calculated as the mean decrease in impurity (Gini Importance).

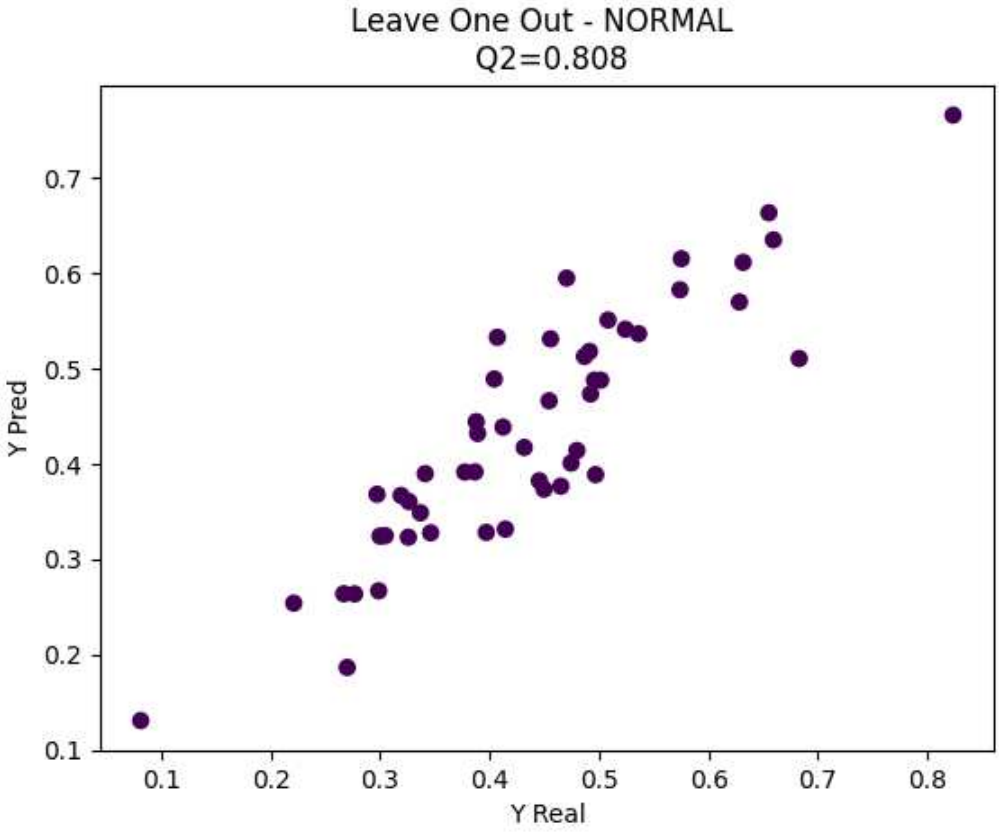

Figure 4. Leave-One-Out (LOO) cross-validation.

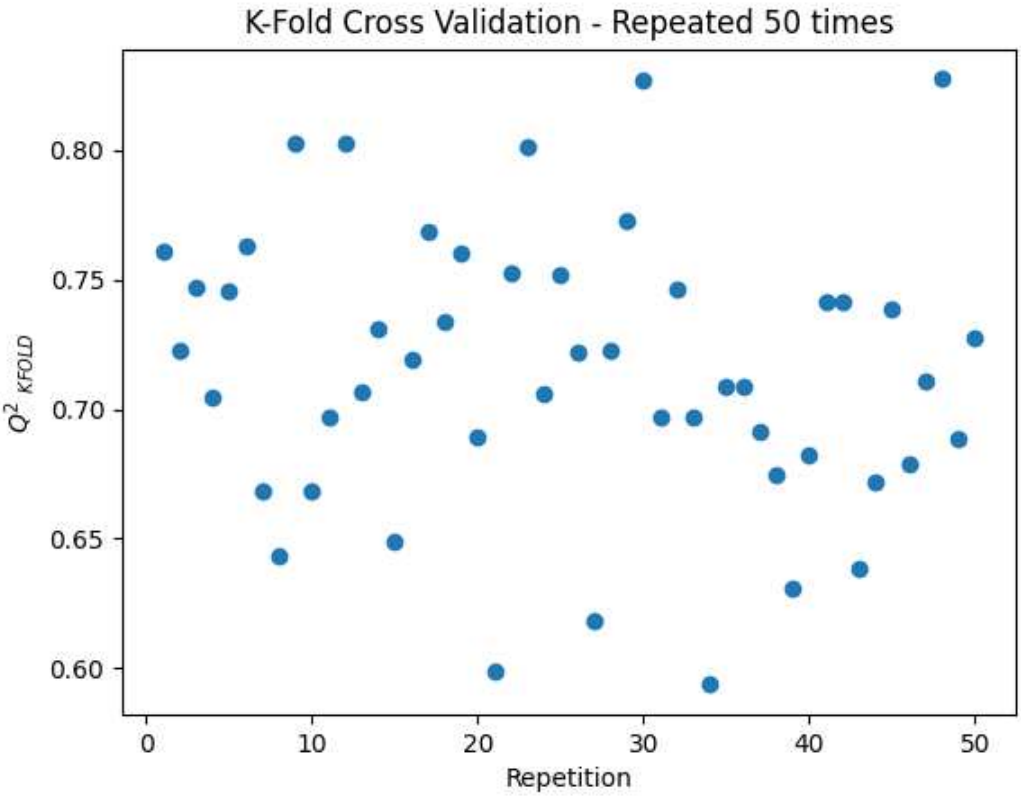

Figure 5. K-Fold cross-validation results (repeated 50 times).

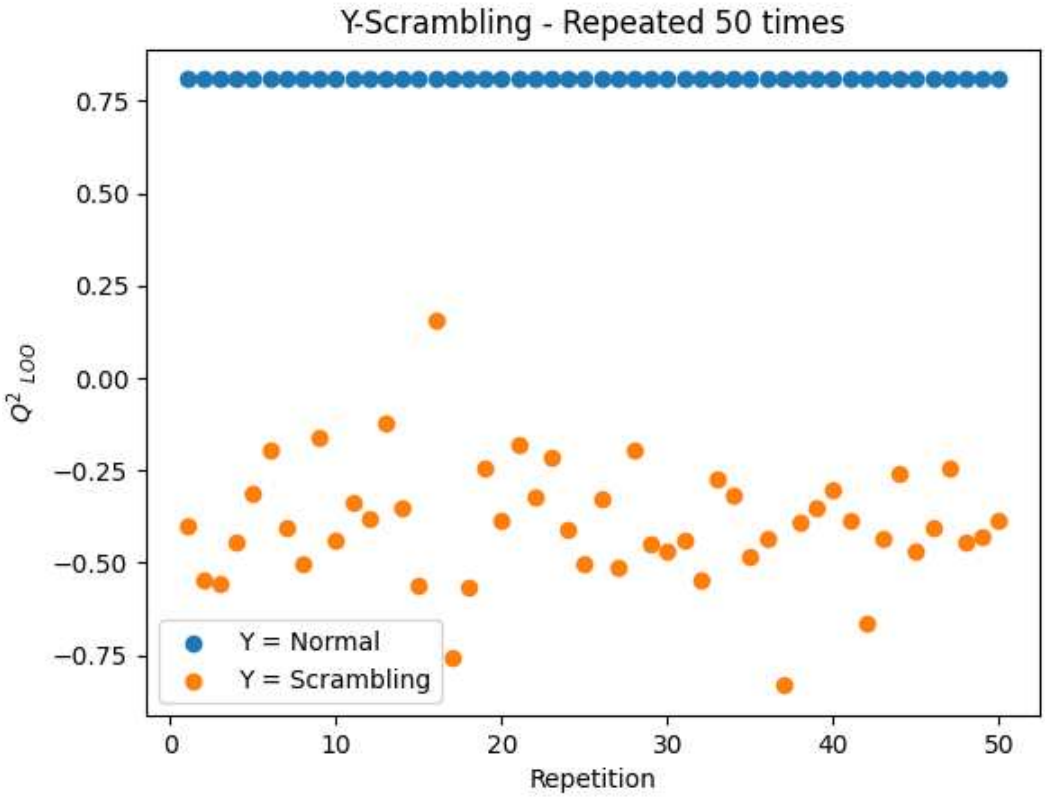

Figure 6. Y-scrambling analysis confirming the absence of chance correlations.

Br. J. Ed., Tech. Soc., v.19, n.1, Jan-Mar, p.185-203, 2026
DOI http://dx.doi.org/10.14571/brajets.v19.n1.185-203
ISSN 2316-9907



Figure 7. User interface to predict values using the trained model.

For a better understanding of the CarcaraQSAR flow, we created a demonstration video showcasing its functionality and key features. The video is available at https://youtu.be/ALBr-ow9J_o.

It is important to consider here that QSAR studies encompass different methodological approaches. Regarding the type of descriptors, they include classical models based on 1D and 2D descriptors, physicochemical properties, as well as 3D-QSAR methods, such as CoMFA and CoMSIA, which explore steric and electrostatic fields to correlate molecular structure and biological activity. In this context, several investigations in Medicinal Chemistry have contributed significantly to the conceptual and practical advancement of QSAR modeling (Avila et al., 2006; Aguirre et al., 2005). These examples highlight the robustness and versatility of in silico approaches in the rational design of bioactive molecules. However, there is no universally superior QSAR method, since the performance and interpretability of the models depend heavily on the nature of the dataset, the chemical space explored, and the quality of the descriptors employed. Thus, different molecular representations — including spreadsheets containing 3D descriptors — can be used as input in platforms such as CarcaraQSAR, if the user wishes, reinforcing the importance of methodological flexibility and the suitability of the model to the chemical-biological problem under study.

3.2 Second example

For a second illustration of CarcaraQSAR, we used a dataset of 55 monoterpenes and structurally related compounds, previously studied by our research group (AllData_SecondExp_CarcaraQSAR.csv) (dos Santos et al., 2018). These compounds had larvicidal activity determined in Aedes aegypti (mosquito transmitting arboviruses) and expressed as concentration in parts per million (ppm) required to promote death in 50% of the population (LC50). Activity data were converted to a logarithmic scale. Classical molecular descriptors were obtained with the Dragon 7 program (Todeschini et al., 2017).

The experiment involved the use of a Bee Colony algorithm for feature selection and Random Forest for both variable selection and final model construction. Figure 8 illustrates the relative importance of the variables, followed by the feature selection and model training, similar to the previous experiment. Of the thousands of variables available, only four were selected after the Bee Colony: ALOGP2, ALOGPS_logP, RDF035u, Xu.

Model validation was also performed using Leave-One-Out (LOO) cross-validation (Figure 9) and K-Fold cross-validation (Figure 10). The results demonstrate a satsfactory model, with a coefficient of determination Q^2_{LOO}=0.679, K-Fold values exceeding 0.5, with minor exceptions. Additionally, the Y-scrambling analysis (Figure 11) confirmed the absence of chance correlations, as all Q^2_{LOO} values for scrambled Y datasets were significantly lower than 0.4.

Similar to the previous model (Figure 7), dispensed with here for economy, an interface is available to add values of new samples to be predicted.

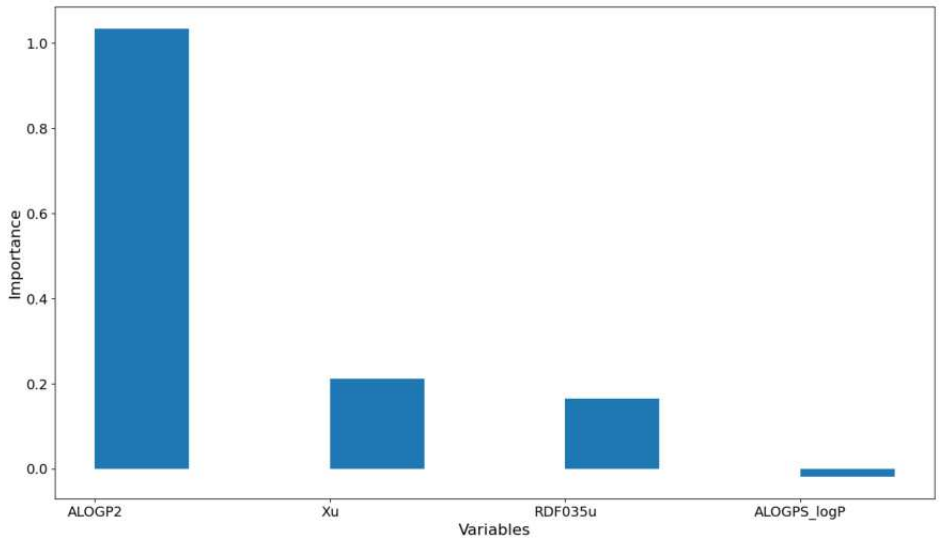

Figure 8. Relative importance of variables after feature selection and model training, calculated as the mean decrease in impurity (Gini Importance).

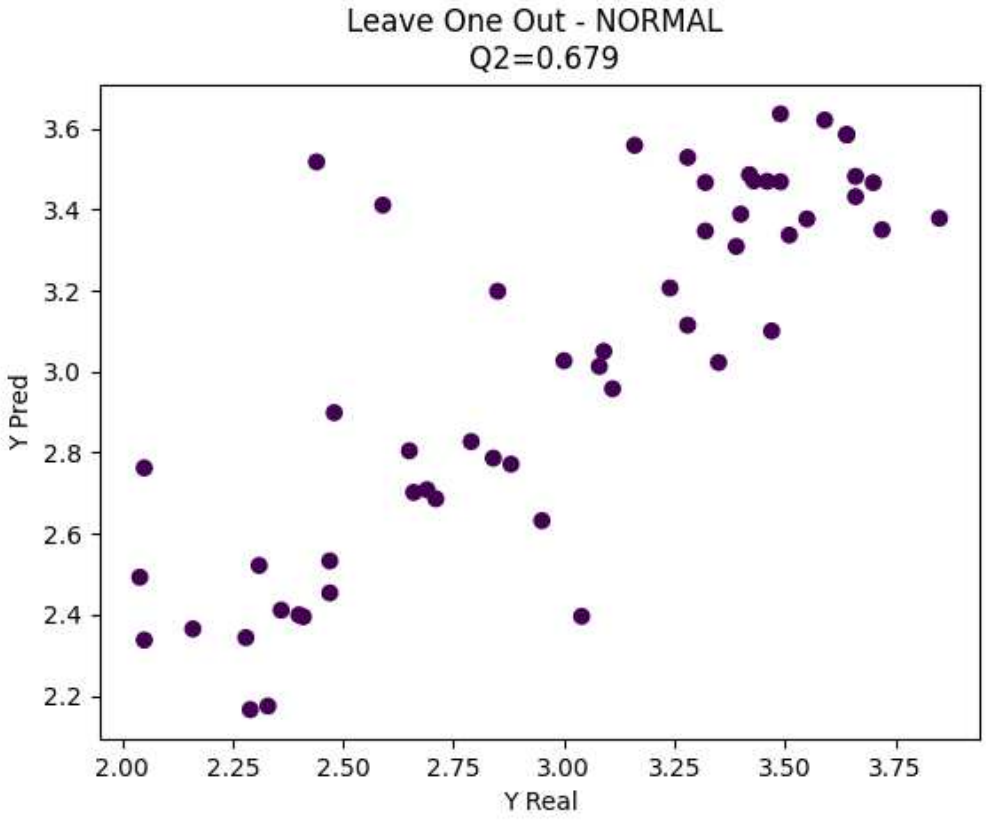

Figure 9. Leave-One-Out (LOO) cross-validation.

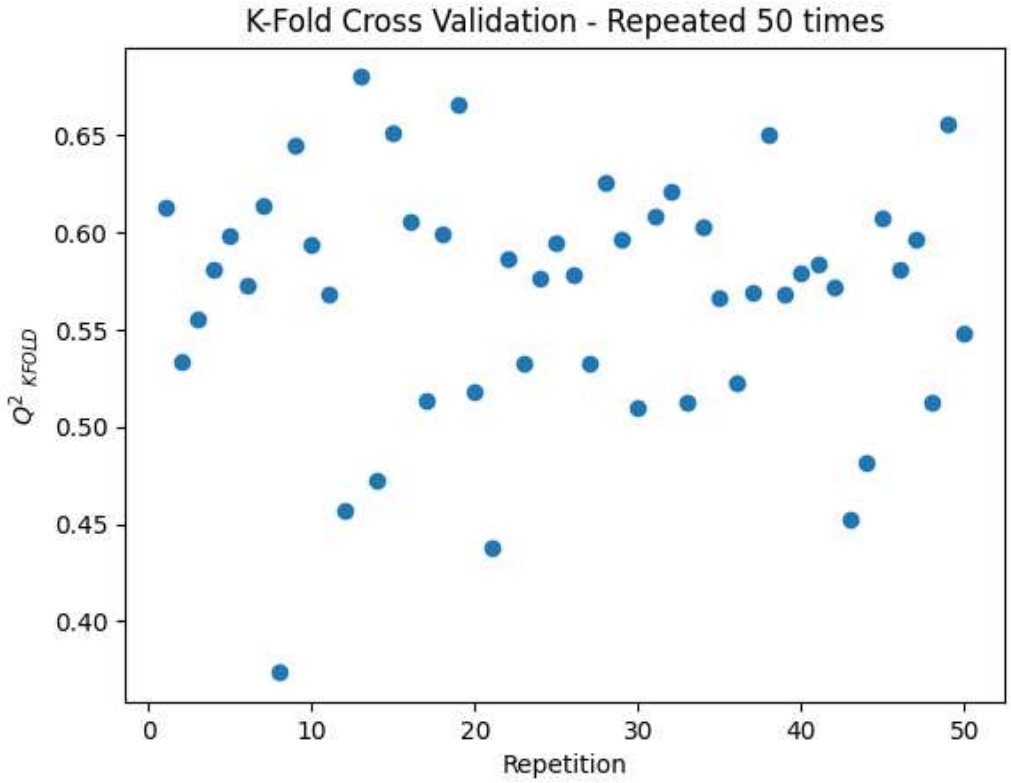

Figure 10. K-Fold cross-validation results (repeated 50 times).

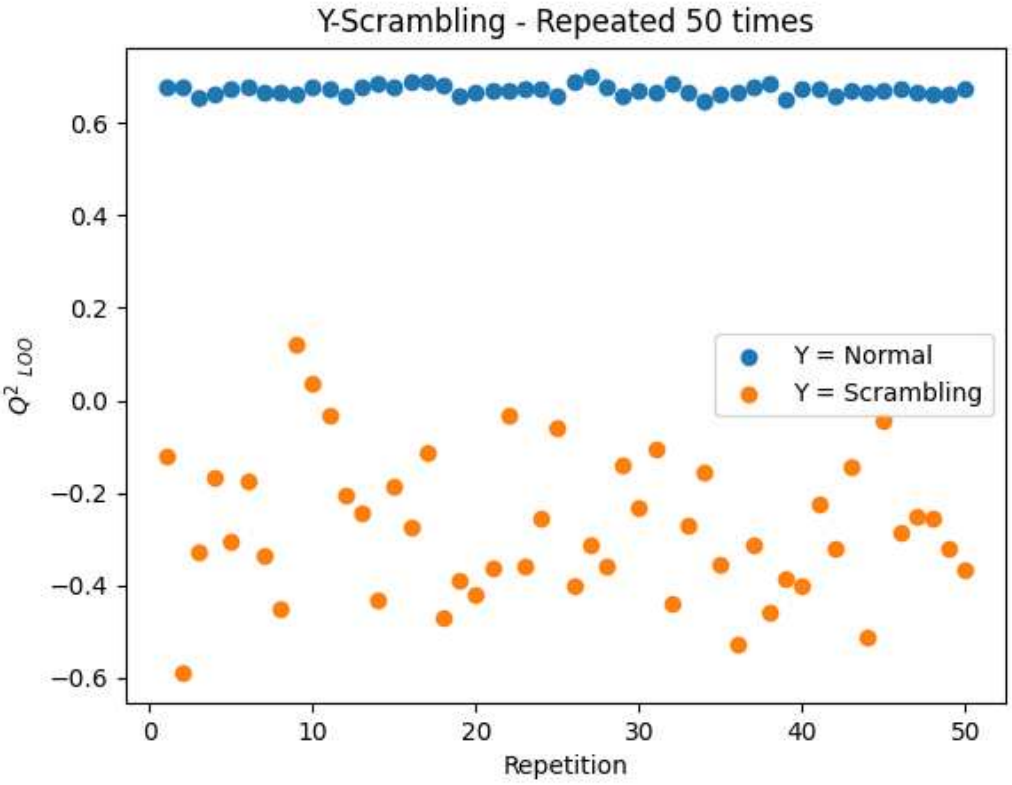

Figure 11. Y-scrambling analysis confirming the absence of chance correlations.

Br. J. Ed., Tech. Soc., v.19, n.1, Jan-Mar, p.185-203, 2026
DOI http://dx.doi.org/10.14571/brajets.v19.n1.185-203
ISSN 2316-9907

3.3 Third example

For the third experiment, we used a dataset of 643 rows and 1025 columns (AllData_ThirdExp_CarcaraQSAR.csv). The molecular SMILES come from the database of Mobley and co-workers (Mobley & Guthrie, 2014), in whose article they provide a compilation of solvation free energies calculated by molecular dynamics or experimentally for a series of molecules. This database would then be useful for building QSPR (Quantitative Structure-Property Relationships) models, in order to predict free energy values. The descriptors were Extended-Connectivity Fingerprints (ECFP), a variant of Morgan fingerprints, calculated by DeepChem, an open-source machine learning framework designed specifically for applications in computational chemistry, bioinformatics, and drug Discovery (Ramsundar et al., 2017) (https://deepchem.readthedocs.io/en/latest/index.html). These are binary vectors that encode the presence or absence of specific chemical substructures in each molecule, generated by means of an algorithm that expands radially from each atom up to a given radius. This approach allows for the compact and efficient representation of the relevant structural characteristics of molecules, and is widely used in machine learning models aimed at predicting physicochemical properties (Ramsundar et al., 2017).

The experiment involved the use of a Bee Colony algorithm for feature selection and Random Forest for both variable selection and final model. 59 variables remained after the bee colony, which makes it impossible to demonstrate the GINI importance graph in a figure.. Figures 12-14 illustrate all validation results, similar to the previous experiments.

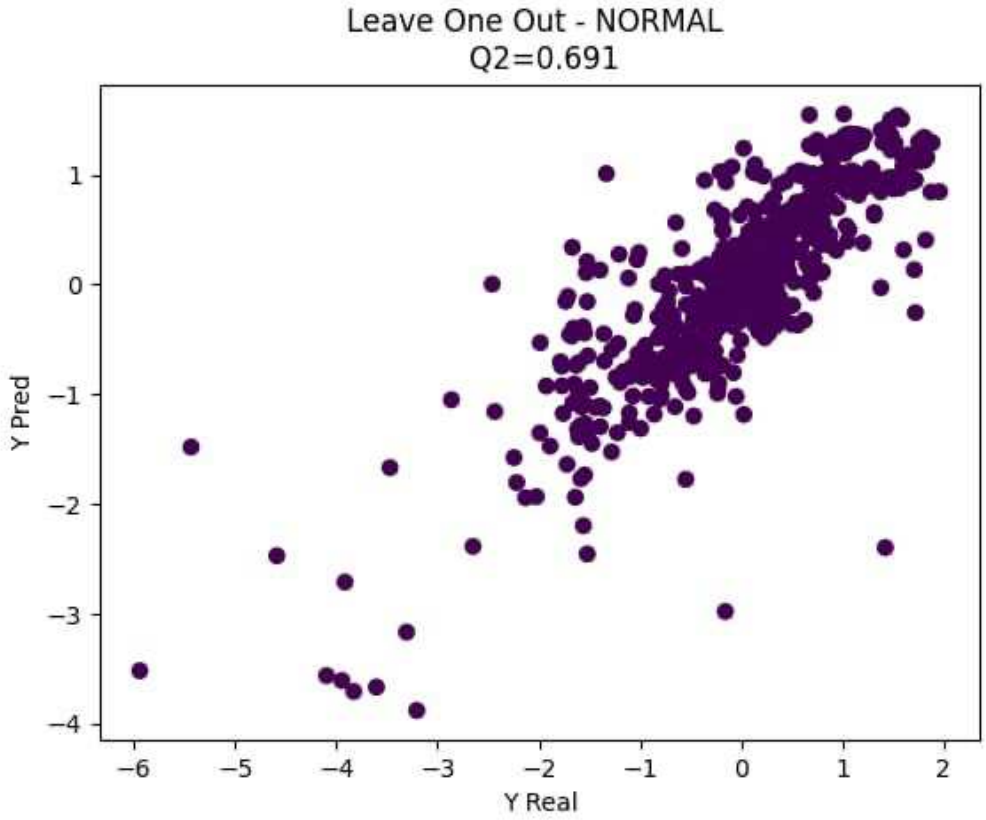

Figure 12. Leave-One-Out (LOO) cross-validation.

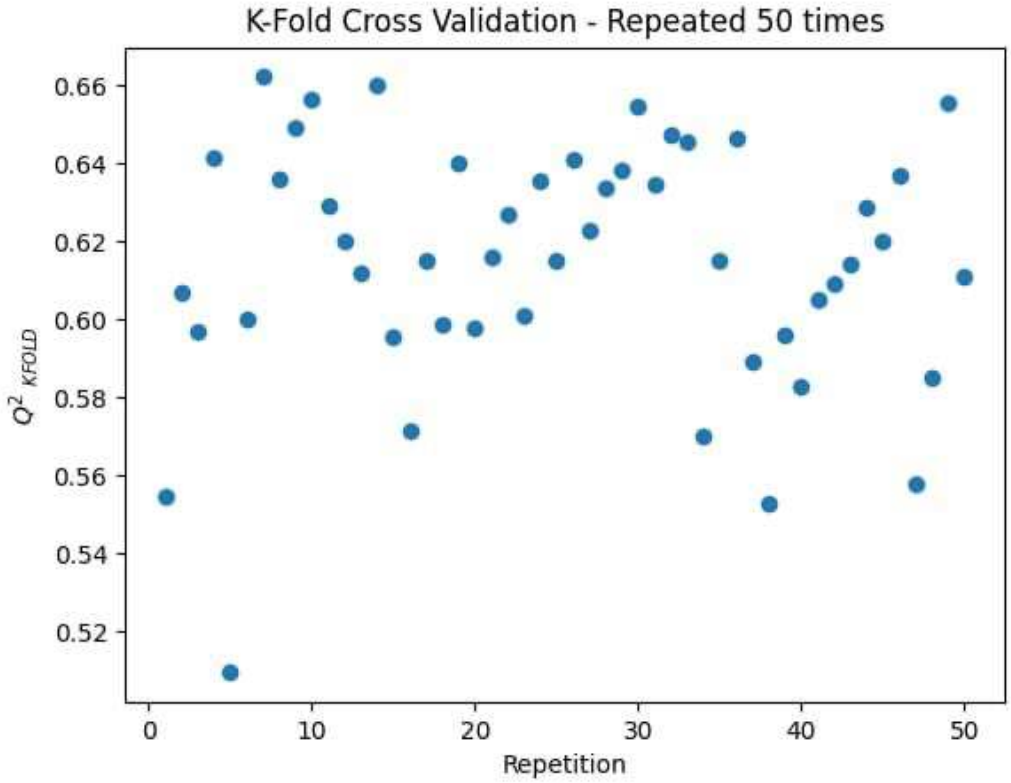

Figure 13. K-Fold cross-validation results (repeated 50 times).

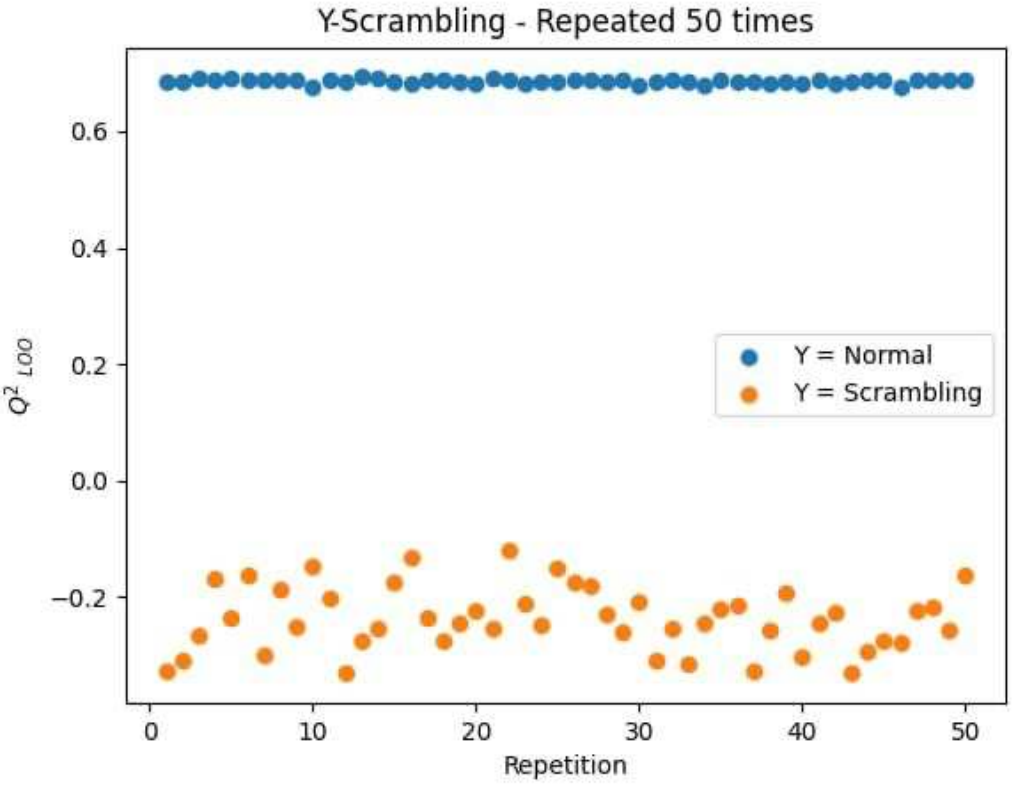

Figure 14. Y-scrambling analysis confirming the absence of chance correlations.

Br. J. Ed., Tech. Soc., v.19, n.1, Jan-Mar, p.185-203, 2026
DOI http://dx.doi.org/10.14571/brajets.v19.n1.185-203
ISSN 2316-9907

3.4 General impact and future directions

CarcaraQSAR provides an accessible and efficient solution for researchers in computational chemistry and pharmaceutical sciences, simplifying the creation of QSAR models. Its user-friendly web-based application removes technical barriers, enabling users without programming expertise to develop accurate predictive models. By integrating advanced machine learning algorithms, such as Random Forest and Support Vector Machines, alongside bioinspired variable selection techniques, the tool improves the reliability and precision of analyses.

The platform's adherence to internationally recognized standards, including the OECD principles for QSAR models, ensures scientific rigor and promotes its use in both academic and industrial setting. CarcaraQSAR provides feedback after model training, helping users refine their models efficiently and make informed decisions throughout the process.

By not relying strictly on a desktop and offering scalability through a cloud-ready design, CarcaraQSAR makes advanced (beyond traditional) computational methods more widely accessible. This contribution supports cost-effective workflows and facilitates progress in drug discovery, molecular research, and education, helping to address these challenges in healthcare, environmental, and related fields.

An important direction for future development is the enhancement of the model selection interface to support automatic discovery and configuration of available machine learning models. We propose implementing a plug-and-play architecture in which the system dynamically detects Python modules containing model definitions and extracts their associated hyperparameters. This improvement would eliminate the need for manual interface updates when new models are added, streamlining the integration process for end users. Such a mechanism would not only facilitate contributions from the community but also simplify the incorporation of new algorithms using generative AI tools such as ChatGPT, further promoting extensibility and adaptability of the CarcaraQSAR platform.

4. CONCLUSION

This paper presented CarcaraQSAR, an open-source, full-stack web application designed to streamline the development of QSAR models. By integrating advanced machine learning techniques, bio-inspired algorithms for feature selection, and adherence to OECD guidelines, CarcaraQSAR simplifies the process of creating robust and reliable predictive models. Its architecture and features, including real-time validation feedback and cloud-ready deployment, were detailed, highlighting its ability to address common barriers such as accessibility, usability, and scalability in computational chemistry.

CarcaraQSAR not only empowers researchers with limited programming expertise but also promotes cost-effective and efficient workflows, enhancing its applicability in diverse scientific and industrial contexts. Its comprehensive feature set offers an innovative approach to QSAR modeling, facilitating advancements in drug discovery, teaching and molecular research.

CarcaraQSAR's main vocation is in relatively small databases, comprising tens to hundreds of molecules, which comprise the majority of the day-to-day reality of synthetic laboratories, natural products or materials science, as well as didactic applications in teaching. For databases of thousands of compounds, on the other hand, application platforms and methods based on deep neural networks such as GNN (Graphic Neural Networks) are recommended, which are also not suitable, on the other hand, for smaller databases.

It is important to highlight that CarcaraQSAR offers valuable contributions to higher education in chemistry, providing a practical tool that complements students' theoretical training, especially in critical disciplines such as pharmaceutical and medicinal chemistry.

Future directions for CarcaraQSAR include extending its capabilities to support more complex machine learning techniques and integrating neural network-based models. Investigating methods to enhance the interpretability of QSAR models and incorporating explainable AI features (as "Large Language Models") could also significantly broaden the tool's impact. These efforts will ensure CarcaraQSAR continues to evolve, addressing emerging challenges and opportunities in Computational Chemistry community.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support provided by the National Council for Scientific and Technological Development (CNPq), which contributed to the successful development of this research.

REFERENCES

Aguirre, G., Boiani, L., Boiani, M., Cerecetto, H., Di Maio, R., González, M., Porcal, W., Denicola, A., Piro, O. E., Castellano, E. E., Sant'Anna, C. M. R., & Barreiro, E. J. (2005). New potent 5-substituted benzofuroxans as inhibitors of *Trypanosoma cruzi* growth: Quantitative structure–activity relationship studies. *Bioorganic & Medicinal Chemistry, 13*, 6336–6346.

Ambure, P., Aher, R. B., Gajewicz, A., Puzyn, T., & Roy, K. (2015). NanoBRIDGES software: Open access tools to perform QSAR and nano-QSAR modeling. *Chemometrics and Intelligent Laboratory Systems, 147*, 1–13. https://doi.org/10.1016/j.chemolab.2015.07.007

Ambure, P., Halder, A. K., González-Díaz, H., & Cordeiro, M. N. D. S. (2019). QSAR-Co: An open source software for developing robust multitasking or multi-target classification-based QSAR models. *Journal of Chemical Information and Modeling, 59*(6), 2538–2544.

Avila, C. M., Romeiro, N. C., Silva, G. M. S., Sant'Anna, C. M. R., Barreiro, E. J., & Fraga, C. A. M. (2006). Development of new CoMFA and CoMSIA 3D-QSAR models for anti-inflammatory phthalimide-containing TNF-alpha modulators. *Bioorganic & Medicinal Chemistry, 14*, 6874–6885.

Bahia, M. S., Kaspi, O., Touitou, M., Binayev, I., Dhail, S., Spiegel, J., & Khazanov, N. (2023). A comparison between 2D and 3D descriptors in QSAR modeling based on bioactive conformations. *Molecular Informatics, 42*(4), e2200196.

Berthold, M. R., Cebron, N., Dill, F., et al. (2009). KNIME – The Konstanz Information Miner: Version 2.0 and beyond. *SIGKDD Explorations Newsletter, 11*(1), 26–31.

Cherkasov, A., Muratov, E. N., Fourches, D., et al. (2014). QSAR modeling: Where have you been? Where are you going to? *Journal of Medicinal Chemistry, 57*(12), 4977–5010.

Dassault Systèmes BIOVIA. (2023). *Discovery Studio Modeling Environment* (Release 2023). San Diego, CA: Dassault Systèmes.

Dos Santos, I. M., Agra, J. P. G., de Carvalho, T. G. C., et al. (2018). Classical and 3D QSAR studies of larvicidal monoterpenes against *Aedes aegypti*: New molecular insights for the rational design of more active compounds. *Structural Chemistry, 29*, 1287–1297.

Gramatica, P., Cassani, S., & Roy, P. P. (2014). QSARINS: A new software for the development, analysis, and validation of QSAR MLR models. *Journal of Computational Chemistry, 35*(13), 1036–1044.

Halder, A. K., & Cordeiro, M. N. D. S. (2021). QSAR-Co-X: An open source toolkit for multitarget QSAR modelling. *Journal of Cheminformatics, 13*, Article 58.

Mauri, A., & Bertola, M. (2022). Alvascience: A new software suite for the QSAR workflow applied to the blood–brain barrier permeability. *International Journal of Molecular Sciences, 23*(21), 12882. https://doi.org/10.3390/ijms232112882

Mobley, D. L., & Guthrie, J. P. (2014). FreeSolv: A database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design, 28*, 711–720.

OECD. (2004). *Principles for the validation, for regulatory purposes, of (quantitative) structure–activity relationship models*. Paris, France: OECD Publishing.

Oliveira Neto, R. F. (2023). *ML-SPARD: A dataset for machine learning performance analysis in small-sample regression problems*. Harvard Dataverse.

Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., & Pande, V. (2017). Is multitask deep learning practical for pharma? *Journal of Chemical Information and Modeling, 57*(8), 2068–2076.

Schrödinger LLC. (2023). *Schrödinger Release 2023-2: QikProp*. New York, NY: Schrödinger LLC.

Supuran, C. T. (2019). Editorial: Teaching medicinal chemistry through computational tools. *Journal of Enzyme Inhibition and Medicinal Chemistry, 34*(1), 1–2.

Sushko, I., Novotarskyi, S., Körner, R., et al. (2011). Online chemical modeling environment (OCHEM): Web platform for data storage, model development and publishing of chemical information. *Journal of Computer-Aided Molecular Design, 25*(6), 533–554.

Todeschini, R., Mauri, A., & Consonni, V. (2017). *DRAGON software (version 7.0) for the calculation of molecular descriptors*. Milano, Italy: Kode Chemometrics srl.

Yap, C. W. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry, 32*(7), 1466–1474.