pyStudio: An Open-Source Machine Learning Platform

Enrique Gomicia-Murcia
Research and Innovation
ELM Company
Riyadh, Saudi Arabia
egomicia@elm.sa

Muhammad AL-Qurishi
Research and Innovation
ELM Company
Riyadh, Saudi Arabia
mualqurishi@elm.sa

Riad Souissi
Research and Innovation
ELM Company
Riyadh, Saudi Arabia
rsouissi@elm.sa

***Abstract*—Data analytics has emerged as a critical capability for businesses and organizations in the modern era. The abundance of data necessitates a deep understanding and the exploitation of its potential to gain insights into current and future scenarios.**

This paper introduces an integrated platform designed to streamline data acquisition, storage, management, processing, and visualization. The primary objective is to facilitate data analysis by offering a machine learning studio equipped with pre-built algorithms. Remarkably, this platform eliminates the need for coding, allowing users to effortlessly generate AI models. Furthermore [9], it provides a secure environment for sharing these models without compromising data privacy—a noteworthy contribution in the realm of federated learning (FL).

The platform's significance lies in its ability to empower non-technical users to perform advanced tasks without requiring specialized expertise.

1. Introduction

The rapid proliferation of artificial intelligence tools has ushered in a new era of data-driven decision-making. These tools continue to grow in number and sophistication, with each advancement adding layers of abstraction to complex mathematical algorithms. However, a fundamental challenge persists – the prerequisite for programming skills to effectively utilize these tools and associated libraries.

Simultaneously, the digital landscape has been reshaped by the consolidation of the big data era [4]. Organizations today find themselves inundated with vast quantities of data, varying in both volume and quality. This wealth of information, intrinsic to businesses, offers the promise of valuable insights into their operations, performance, and potential future directions.

Yet, a stark problem arises - users without technical skills often grapple with the complexity of handling this data. For them, using artificial intelligence tools is either a daunting task or an impossibility, depriving them of the inherent value and knowledge locked within their data.

While commercial platforms from tech giants like Microsoft and IBM exist, their accessibility is often limited by prohibitive costs. Furthermore, these platforms are typically off-the-shelf solutions, challenging businesses with unique use cases or specialized needs. They lack the flexibility for customization or tailoring to specific business requirements.

In response to this predicament, we present an innovative open-source platform. Our platform opens the door for non-technical users to harness the power of artificial intelligence tools, enabling them to share, explore, and exploit their data for valuable insights, all without the need for programming or querying languages. It empowers users through the simplicity of visual programming, democratizing data analytics for all. we have developed a user-friendly visual programming platform. It simplifies coding by allowing users to select visual elements representing common programming instructions like IF, ELSE, FOR, WHILE, and more. Users can manage variables and create functions using graphical blocks, eliminating the need for traditional coding.

Being open-source, our platform offers the additional advantage of customization and extensibility. Users can tailor the platform to meet their unique needs, from its appearance to the algorithms it employs, all without incurring additional costs or external support.

To enhance user experience and usability, our platform features a unified web application interface. This interface provides an intuitive means of managing data and working with machine learning tools. Users can navigate through four primary features: the Dashboard for project and data management, the Marketplace for publishing and discovering data sets and AI models, Notebooks for technical users, and the Machine Learning studio for creating workflows through drag-and-drop machine learning algorithms. For administrators, a fifth feature provides control over user management and permissions.

These user-centric features are underpinned by a robust system architecture, depicted in Figure 1, which we will elaborate upon in the following section of this paper.

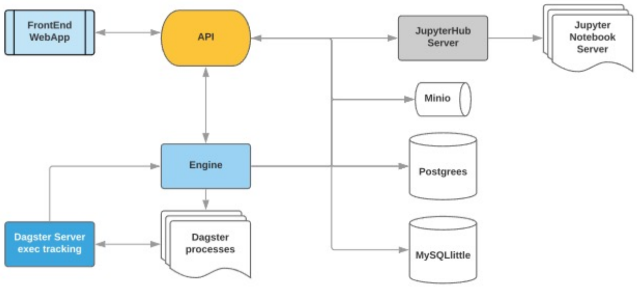


Figure 1. General system architecture.

1.1. System Architecture

When developing our platform, we had used a Microservices architecture, which is a modern interpretation of service-oriented architecture focusing on distributing functionality across discrete services, instead of having just one application with a monolithic design. Our services communicate with each other through the main service, which is an API that conforms to the design principles of the REST (Representational State Transfer) architectural style. Using REST as the communication technology allows us to employ different programming languages and frameworks, facilitating continuous deployment [5], integration, and other agile practices.

1.1.1. Front-End Service. Our Front-End service, responsible for running the web application, is built on the Vue.js framework and utilizes HTML and CSS. It incorporates styling elements from Vuetify and Bootstrap to enhance the user interface. Within the Front-End, the most intricate component is the Machine Learning Studio, which leverages a suite of technologies, including JsPlumb, JQuery, JQuery-UI, and Html2pdf on the JavaScript side.

- **JsPlumb:** Manages the connections between machine learning nodes.
- **JQuery-UI:** Facilitates drag-and-drop interactions between nodes.
- **JQuery:** Simplifies JavaScript coding.
- **Html2pdf:** Enables the generation of PDF files.

Vuetify and Bootstrap contribute to the aesthetic appeal of the user interface, encompassing elements such as inputs, buttons, and animations.

1.1.2. API Service. Our primary service, known as API, serves as the orchestrator of all communications between the Front-End and other services. This pivotal service handles various critical functions, including:

- **Security:** Implementing OAuth authentication methods.
- **User Management:** Utilizing SQLittle as a storage solution.
- **Proxying:** Acting as a proxy for the Engine service, JupyterHub-Server, and MINIO object storage.

To realize these functionalities, we have chosen the Django framework. Django, a high-level Python web framework, is renowned for its capacity to facilitate rapid development and uphold clean, pragmatic design principles. It is an open-source framework that effectively manages the intricacies of web development.

1.1.3. Engine Service. The core of our machine learning studio is the Engine service, constructed using Flask, a Python web framework known for its minimal core and extensibility. Within the Engine service, we implement each of the droppable functionalities, governing the execution and management of workflows. To further enhance our capabilities, we have integrated Dagster, an orchestrator designed explicitly for the development and maintenance of data assets. These assets encompass tables, data sets, machine learning models, and reports. To support the execution of Dagster, our Engine service relies on Postgres DB.

1.1.4. JupyterHub-Server Integration. In our quest to provide an easy user experience, we have integrated JupyterHub-Server into our platform. This multi-user Hub efficiently spawns, manages, and proxies multiple instances of the single-user Jupyter notebook server. The integration is transparent to the user, creating the illusion of a unified web application experience.

2. Main Features

Our platform offers a comprehensive set of features designed to empower data scientists and AI practitioners at every stage of their projects. These features enhance and facilitate machine learning tasks.

2.1. Dashboard for Project and Data Management

The Dashboard serves as the central hub for AI models and data management. It simplifies organization, enabling users to track progress effortlessly. Additionally, it acts as a centralized repository for their datasets and models, ensuring easy access and allowing them to publish them easily making them visible on the marketplace.

Users can access and utilize their deployed models conveniently from the Dashboard. They can perform single-call inference by specifying parameters and invoking models with a single click. Additionally, bulk inference is supported, allowing users to process CSV files using their previously deployed models.

2.2. Marketplace for Data Sets and AI Models

Our platform hosts a dynamic Marketplace that connects users with a wealth of datasets and pre-trained AI models. Here, users can discover a wide range of resources, share their own datasets and models, and collaborate with peers on data-driven projects.

2.3. Notebooks for Technical Users

Technical users can leverage our powerful Notebook environment, offering support for various programming languages and libraries. This interactive coding space facilitates data exploration, analysis, and visualization, making it an ideal tool for in-depth analysis.

2.4. Machine Learning Studio

Our Machine Learning Studio is the heart of our platform, for Model Development and Deployment, catering to users of all technical backgrounds. It boasts a user-friendly interface that enables both beginners and experts to effortlessly create machine learning models and build intricate workflows. Offering integration of data from various sources [3]. Users can choose from existing datasets in their dashboard, upload CSV files, or access Kaggle datasets through our integrated API, simplifying data utilization.

Notably, it streamlines model deployment [5], allowing users to convert their models into web services directly from the Studio an effortless devOps [6]. This capability ensures real-time predictions and improves overall accessibility for end-users. Details about the various available algorithms will be explained in the following section.

2.5. Administrative Tools

For administrators, our platform provides essential tools to manage user roles and permissions effectively. These tools ensure security, access control, and the overall well-being of the platform, allowing for smooth operation.

3. Supported Machine Learning Algorithms

In this section, we provide a brief overview of the diverse range of algorithms supported by our open-source platform [10]. We believe in empowering our users with flexibility and choice, allowing you to incorporate the algorithms that best suit your machine learning needs. You have the freedom to explore and extend these algorithms, as many of them are integral parts of the renowned scikit-learn library.

Our platform offers a comprehensive selection of algorithms [2] that can be broadly categorized into three main groups: Incremental Learning, Top Machine Learning Algorithms, and Classic Machine Learning Algorithms.

At main level of all groups you will find two fundamental functions. ”Fit Model” algorithm serves as a valuable tool. It allows users to input a labeled column and obtain a fitted (trained) model as the output. On the other hand, the ”Run Model” algorithm is designed for making predictions with a fitted model. Simply specify an unlabeled column, and it will generate predictions for the unlabeled data.

Within these categories, we present more than twenty algorithms, each accompanied by a brief description of its purpose and functionality. The provided basic parameters serve as a starting point for your machine learning journey. Remember, these parameters can be expanded and fine-tuned to align with your specific requirements.

3.1. Incremental Learning Algorithms

Our platform offers a range of Incremental Learning algorithms, designed to facilitate continuous model updates. The ”Load Model” algorithm allows users to retrain models, where they specify the ”Model Name” parameter to choose the model for retraining.

- **Classification Algorithms**: In the realm of Classification, our platform provides several algorithms. The ”SGDC Classifier” algorithm lets users set the learning rate parameter. Algorithms like ”MultinomialNB” and ”BernoulliNB” offer control over parameters like ”Alpha” and ”Fit Prior.” The ”Perceptron” algorithm provides flexibility with parameters such as ”Penalty,” ”Alpha,” and ”Fit Intercept.” Lastly, the ”Passive Aggressive Classifier” enables fine-tuning with the ”Step Size” and ”Fit Intercept” parameters.
- **Regression Algorithms**: For Regression tasks, our platform supports the ”Passive Aggressive Regressor” algorithm, allowing users to adjust its behavior using parameters such as ”Step Size,” ”Epsilon,” and ”Fit Intercept.” These parameters empower users to tailor the regression model’s performance to their specific requirements.
- **Clustering Algorithms**: In the realm of Clustering, our platform offers the ”MiniBatchKMeans” algorithm. Users can customize the clustering process by specifying parameters like the ”Number of Clusters,” ”Number of Iterations,” and whether to ”Fit Intercept.” These parameters provide control over the clustering algorithm, ensuring it aligns with various data structures and analysis needs.

3.2. Top Algorithms

The top algorithms, which are frequently employed in Kaggle competitions and machine learning tasks, encompass a diverse range of models designed for both classification and regression tasks.

- **XGBoost (Extreme Gradient Boosting)**: XGBoost is a powerful gradient boosting algorithm suitable for both classification and regression tasks. It offers parameters like ”Tree Method” for specifying the tree construction method.
- **LightGBM**: LightGBM, another gradient boosting framework, is celebrated for its speed and efficiency. It employs a histogram-based learning method, which accelerates training by discretizing continuous features into discrete bins. Key parameters in LightGBM include the number of leaves, learning rate, and the maximum bin size.
- **CatBoost (Categorical Boosting)**: CatBoost specializes in handling categorical features effectively. It is particularly suitable for tabular data and offers features like ordered boosting, which optimizes the

boosting order to reduce overfitting. Parameters in CatBoost include the number of iterations, task type (CPU or GPU), and the depth of the trees.

- **BaggingClassifier**: BaggingClassifier is an ensemble method that combines multiple classifiers to improve classification accuracy. Users can fine-tune it with parameters such as ”Random State” and ”Max Features.”
- **LGBMRegressor (LightGBM Regressor)**: LGBMRegressor, based on LightGBM, is designed for regression tasks. It provides parameters like ”Number of Leaves,” ”Learning Rate,” and ”Number of Estimators” for optimization.
- **Adaboost**: Adaboost is an ensemble method that combines weak classifiers to create a strong classifier. Users can customize it with parameters like the ”Algorithm” choice.
- **LSTM Model (Long Short-Term Memory)**: LSTM Model is a deep learning model suitable for sequence prediction tasks. It offers parameters like the number of training ”Epochs” for fine-tuning.

These top algorithms provide data scientists with versatile tools for tackling a wide range of machine learning problems, from tabular data classification to regression tasks. By carefully adjusting their parameters, practitioners can achieve remarkable results and win Kaggle competitions.

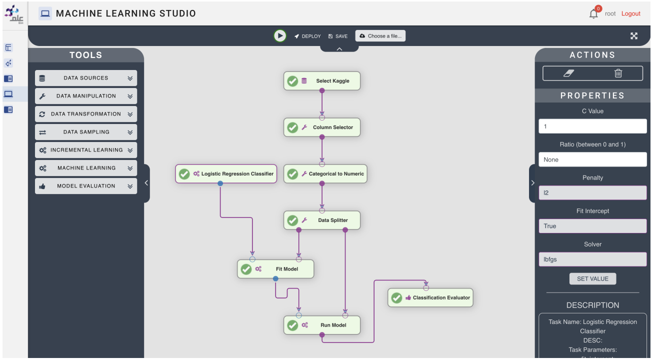
3.3. Classic Algorithms Supported

In this section, we introduce a set of classic machine learning algorithms that are commonly used in various data science and machine learning tasks [8]. These algorithms are well-established and provide a solid foundation for building predictive models and performing data analysis. Below, we provide brief descriptions of each algorithm along with key parameters that users can customize to suit their specific needs.

- **Random Forest Classifier:** A versatile ensemble method for classification tasks. Users can set the number of trees in the forest (`n_estimators`) to control model complexity.
- **Linear Regression:** A fundamental algorithm for regression tasks. Users can choose to normalize the input features (`normalize`) for improved model performance.
- **Random Forest Regression:** For regression tasks using random forests. Parameters include `n_estimators`, `criterion`, `min_samples_split`, `min_samples_leaf`, and `random_state`.
- **Decision Tree Classifier:** A simple yet powerful classification algorithm. Users can customize parameters such as `criterion`, `min_samples_split`, `min_samples_leaf`, `max_depth`, `max_leaf_nodes`, and `random_state`.
- **k-Nearest Neighbor Classifier:** A simple classification algorithm based on nearest neighbors. Users can set the number of neighbors (`n_neighbors`) to influence the prediction.
- **Logistic Regression Classifier:** Widely used for binary and multiclass classification. Users can adjust parameters such as `C`, `penalty`, `fit_intercept`, and `solver`.
- **Multi-layer Perceptron Classifier:** A neural network for complex classification. Users can customize parameters including `activation`, `alpha`, `hidden_layer_sizes`, `learning_rate`, `learning_rate_init`, `max_iter`, `random_state`, and `solver`.
- **k-Means Cluster:** An unsupervised clustering algorithm. Parameters include `n_clusters`, `init`, `n_init`, `max_iter`, and `tolerance`.

4. Demo

Utilizing the Kaggle Breast Cancer dataset [7], we swiftly established a machine learning workflow, as illustrated in Figure 2. We initiated this process by easily selecting the desired columns with a few simple mouse clicks. Subsequently, we seamlessly integrated a ’categorical to numerical’ data transformation algorithm into our workflow, effortlessly selecting the relevant column for transformation. Finally, we executed the customary machine learning procedures, including data splitting, model fitting, and execution.



C Value: 1
Ratio (between 0 and 1): None
Penalty: l2
Fit Intercept: True
Solver: lbfgs

Task Name: Logistic Regression Classifier
DESC:
Task Parameters:

Figure 2. Initial workflow demonstration.

Thanks to our user-friendly platform constructing the initial workflow required a mere minute, and transitioning between four distinct algorithms, as previously mentioned, took only a few seconds. Given that we are addressing a classification task, we opted for the classic Logistic Regression alongside the commonly employed Adaboost, XGBoost, and Bagging Classifier. This showcases how users can effortlessly switch between algorithms, enabling them to evaluate and determine the most suitable algorithm for their specific use case. On Table 1 we present a comprehensive comparison of the outcomes obtained from these diverse models, emphasizing the ease and speed at which users can perform such evaluations on our platform.

TABLE 1. Model Comparison on Breast Cancer Dataset

Model	Parameters Used	Accuracy	Precision	Recall	F1 Score
Logistic Regression	C: 1, Ratio: None, Penalty: l2, Fit Intercept: True, Solver: lbfgs	0.94	0.82	0.98	0.96
XGBoost Classifier	Using gbTree	0.95	0.85	0.95	0.97
Bagging Classifier	Random State: 10, Max Features: 1	0.94	0.81	0.94	0.96
AdaBoost	Algorithm: SAMME.R	0.98	0.92	0.98	0.98

5. Conclusion

In this demonstration paper, we have introduced our comprehensive AI platform, designed to address the evolving needs of data scientists, developers, and organizations in the field of machine learning. Starting from an abstract that laid the foundation for our platform's significance, we journeyed through its main features.

Our platform embodies innovation at every turn, from the intuitive Dashboard for project and data management to the cutting-edge Marketplace for data sets and AI models. It bridges the gap between technical and non-technical users with its versatile Notebooks and empowers them with the drag-and-drop capabilities of the Machine Learning Studio.

One of the platform's standout features is its commitment to open-source principles. With a powerful GPL v3 license, our platform not only empowers users but invites collaboration from the global community of AI enthusiasts and experts. This open philosophy fosters a collaborative environment, encouraging the sharing of knowledge, expertise, and code for the greater benefit of all.

Moreover, the platform seamlessly integrates multiple data sources, including users' existing data, CSV files, and Kaggle datasets via API integration. This democratization of data access is a testament to our mission of making AI accessible to all.

Additionally, the platform simplifies model deployment, enabling users to effortlessly transform their machine learning models into accessible web services. This ensures that the benefits of AI can be realized in real-time, enhancing accessibility for end-users and organizations alike.

In summary, our platform represents a pioneering effort to democratize AI and machine learning through open-source collaboration. By connecting people, data, and models, it fosters a collaborative environment that drives innovation and empowers organizations to harness the full potential of AI. We believe that our platform is not just a tool but a catalyst for the next wave of AI-driven transformations across industries.

As we move forward, our commitment remains unwavering - to continue enhancing and expanding our platform, and to work in tandem with the AI community to shape the future of machine learning. We invite you to explore our platform firsthand, join us on this exciting journey, and contribute to the open-source revolution that is at its core.

Acknowledgments

The authors would like to express their gratitude to their employer, elm.sa, for supporting and encouraging contributions to the open-source community. This project represents our commitment to advancing technology and fostering collaboration in the field of artificial intelligence.

We also appreciate the contributions of the entire team at elm.sa who have provided valuable feedback and assistance during the development process.

References

[1] https://github.com/elmpystudio Riyadh, Saudi Arabia.

[2] Anamaria Mojica-Hanke, Andrea Bayona, Mario Linares-Vásquez, Steffen Herbold, and Fabio A. González, "What are the Machine Learning best practices reported by practitioners on Stack Exchange?" *arXiv preprint arXiv:2301.10516*, 2023.

[3] Petar Radanliev and David De Roure, "New and emerging forms of data and technologies: Literature and bibliometric review," *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 2887–2911, 2023.

[4] CPA John Kimani and James Scott, "Advanced Big Data Analytics Professional Level," *Finstock Evarsity Publishers*, 2023.

[5] Efterpi Paraskevoulakou and Dimosthenis Kyriazis, "ML-FaaS: Towards exploiting the serverless paradigm to facilitate Machine Learning Functions as a Service," *IEEE Transactions on Network and Service Management*, 2023.

[6] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," *IEEE Access*, 2023.

[7] Selina Sharmin, Tanvir Ahammad, Md Alamin Talukder, and Partho Ghose, "A hybrid dependable deep feature extraction and ensemble-based machine learning approach for breast cancer detection," *IEEE Access*, 2023.

[8] M. Dhinu Lal and Ramesh Varadarajan, "A review of machine learning approaches in synchrophasor technology," *IEEE Access*, 2023.

[9] Fida K. Dankar and Nisha Madathil, "Using Synthetic Data to Reduce Model Convergence Time in Federated Learning," *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 293–297, 2022.

[10] Migran N. Gevorkyan, Anastasia V. Demidova, Tatiana S. Demidova, and Anton A. Sobolev, "Review and comparative analysis of machine learning libraries for machine learning," *Discrete and Continuous Models and Applied Computational Science*, vol. 27, no. 4, pp. 305–315, 2019.