

# Integration of web scraping, fine-tuning, and data enrichment in a continuous monitoring context via large language model operations

Anas Bodor<sup>1</sup>, Meriem Hnida<sup>1,2</sup>, Najima Daoudi<sup>1,3</sup>

<sup>1</sup>ITQAN Team, LyRica Lab, Information Sciences School, Rabat, Morocco

<sup>2</sup>RIME Team, Mohammadia School of Engineers, Mohammed V University, Rabat, Morocco

<sup>3</sup>SSLab, ENSIAS, Mohammed V University, Rabat, Morocco

## Article Info

### Article history:

Received May 23, 2024

Revised Sep 16, 2024

Accepted Oct 1, 2024

### Keywords:

Continuous monitoring

Data enrichment

Fine-tuning

LLMOps

MLOps

Web scraping

## ABSTRACT

This paper presents and discusses a framework that leverages large-scale language models (LLMs) for data enrichment and continuous monitoring emphasizing its essential role in optimizing the performance of deployed models. It introduces a comprehensive large language model operations (LLMOps) methodology based on continuous monitoring and continuous improvement of the data, the primary determinant of the model, in order to optimize the prediction of a given phenomenon. To this end, first we examine the use of real-time web scraping using tools such as Kafka and Spark Streaming for data acquisition and processing. In addition, we explore the integration of LLMOps for complete lifecycle management of machine learning models. Focusing on continuous monitoring and improvement, we highlight the importance of this approach for ensuring optimal performance of deployed models based on data and machine learning (ML) model monitoring. We also illustrate this methodology through a case study based on real data from several real estate listing sites, demonstrating how MLflow can be integrated into an LLMOps pipeline to guarantee complete development traceability, proactive detection of performance degradations and effective model lifecycle management.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Anas Bodor

ITQAN Team, LyRica Lab, Information Sciences School

Rabat, Morocco

Email: anas.bodor@esi.ac.ma

## 1. INTRODUCTION

Machine learning operations (MLOps), stands for a methodology for efficiently managing the development, deployment and management processes of machine learning (ML) models, and large language model operations (LLMOps) [1], which extends these principles specifically to language models such as generative pre-trained transformer (GPT), taking into account their unique requirements [2], combining continuous monitoring [3], model explicability and systematic management of the ML model lifecycle, provide a comprehensive solution for optimizing the performance of deployed models. This approach relies on the use of advanced technologies such as real-time web scraping, Kafka and Spark Streaming for efficient data acquisition and processing. The challenge of continuous monitoring lies in ensuring the reliability and performance of ML models in production. This entails constant monitoring of the results generated by these models, as well as the proactive detection of performance drifts or undesirable behaviors. By placing continuous monitoring and model explicability at the heart of our methodology, we aim to guarantee the

reliability of predictions, while meeting the needs of other application domains. This approach thus provides a solid foundation for informed decision-making, essential in a context where data plays an increasingly crucial role in strategic choices.

This paper introduces an innovative framework that combines MLOps and LLMOps to improve ML models through data enrichment and continuous monitoring. For instance, the framework uses large language models (LLMs) for data enrichment of real-time data acquisition and processing to facilitate predictive accuracy and efficiency. We further validate the framework's effectiveness through a case study in the real estate sector, showcasing its capacity to refine model predictions across various fields. Moreover, our framework ensures the performance of deployed models via rigorous monitoring and lifecycle management, setting a new standard for informed decision-making and significantly advancing the machine learning operations domain.

The remainder of this article is organized as section 2 reviews related works, situating our approach within the existing landscape of MLOps, LLMOps, and continuous monitoring practices. Section 3 introduces our theoretical framework, detailing the underpinnings of LLM for data enrichment, LLMOps for managing the lifecycle of large language models, and the critical role of continuous monitoring in model optimization. Section 4 describes the proposed framework for dynamic optimization of ML models, outlining our multi-step methodology that includes data acquisition, preprocessing, exploration, model building, evaluation, and deployment with continuous monitoring. Section 5 presents a case study to demonstrate the practical application and effectiveness of our framework in improving real estate price prediction models. Finally, section 6 concludes the article by summarizing our findings and discussing the implications of our research for the field of machine learning operations in continuously changing data environments.

## 2. RELATED WORKS

In the context of artificial intelligence (AI), continuous model monitoring plays a vital role. This process involves constant observation of the performance of deployed models to quickly identify any deterioration in the accuracy or reliability of predictions [4]. However, the success of this monitoring is closely dependent on the quality of the underlying data, which underlines the importance of data quality. Poor-quality data can compromise the efficiency and validity of model results [5]. Furthermore, MLOps practices [6], aimed at optimizing the deployment and management of machine learning models, as well as LLMOps practices [7] applied to LLM models [8], [9] dealing with natural language processing (NLP) [10], [11], are also crucial in this context to bring development and production environments even closer together. Integrating these aspects helps maintain model quality and performance [12], which is essential in an environment where data and conditions can change rapidly [13]. Thus, understanding the relationship between continuous monitoring, data quality, MLOps and LLMOps practices is essential to ensure reliable predictions and informed decision-making in diverse application domains.

Several studies have highlighted the current ecosystem of tools that support the ML pipeline. These tools play an essential role in the effective implementation of continuous monitoring, data quality, MLOps practices and LLMOps. Their availability and use enable teams to develop and deploy artificial intelligence models faster and more reliably. For example, in the field of continuous monitoring, tools such as Prometheus [14], Grafana [14] and tensor board [15] provide advanced features for monitoring model performance in real time. These tools enable teams to closely monitor key metrics [16] such as precision, recall and F-score, and quickly detect any deviation from predefined thresholds.

When it comes to data quality, tools such as Great Expectations [17], DataRobot [18], and Trifacta [19] offer features to evaluate, clean and validate data before it is used in ML models. These tools identify outliers, duplicates, missing values and inconsistencies in datasets, helping to improve the quality and reliability of model predictions. In the area of MLOps and LLMOps practices, platforms such as Kubeflow [20], ML flow [21] and Seldon core [22] provide functionality to automate and orchestrate the deployment, management and monitoring of ML models. These platforms enable teams to collaborate efficiently, track the evolution of models and guarantee their consistency and reliability in production environments. By understanding the landscape of available tools, teams can choose the solutions that best meet their specific needs, and implement robust processes to guarantee the quality and reliability of artificial intelligence model predictions.

## 3. THEORETICAL FRAMEWORK

Before exploring the details of the framework that integrates LLM and ML models with real-time data processing, as outlined in section 4, it is important to first establish its relevance within the broader context of our research. This framework is not only a technical contribution but also addresses critical gaps in

current methodologies by offering a novel approach to handling dynamic data. To fully appreciate the significance of this integration, key concepts related to both machine learning and natural language processing, as well as their roles in modern data-driven environments, need to be understood. These foundational ideas will help highlight the importance of our contribution to advancing research in this field.

### 3.1. LLM for data enrichment

The LLM approach is important for extracting information from textual descriptions due to its contextual understanding capabilities. LLMs [23] like GPT, bidirectional encoder representations from transformers (BERT) and Bard [24], [25] are trained on extensive text datasets, enabling them to grasp the context of a given textual description. This contextual comprehension allows them to capture implicit meanings, which is crucial for precise information extraction. LLMs can efficiently process large volumes of text across a wide range of domains and subjects. They can be refined or customized for specific information extraction tasks. By providing additional training data or specific instructions, users can adapt the model to extract desired information more accurately for particular applications or domains. These qualities make LLMs valuable tools for a wide range of applications, from natural language understanding to knowledge management. LLMs are used in various language-related applications [26]:

- a. Automatic translation [27]: LLMs can translate texts from one language to another with impressive accuracy.
- b. Text generation [28]: LLMs can generate blog articles, summaries, or product descriptions based on a set of keywords or input text.
- c. Question answering [29]: LLMs can provide accurate answers to complex questions based on the information available in the input texts.
- d. Intelligent personal assistant [30]: LLMs can function as conversational agents to assist users in various tasks such as note-taking, information retrieval, or event planning.

Data enrichment is a crucial process that enhances the value of data by refining and augmenting them with additional attributes. By enriching data, we gain deeper insights into our dataset. Data enrichment involves several common tasks, including adding data, segmentation, deriving attributes, data imputation, entity extraction, and data categorization. In this context, NLP [31] and machine learning techniques are often employed, particularly with models such as language model for data enrichment (LLM), which automate and enhance these processes.

### 3.2. LLMOps for automating LLMs lifecycle

LLMOps is an emerging methodology that aims to streamline and automate the lifecycle of LLMs in production, as this type of ML model can generate results in human language. It is a specialization of MLOps adapted to the specific challenges of LLMs. LLMOps focuses specifically on the lifecycle management of large language models, such as those used in automatic NLP. It includes specialized tools and practices tailored to the unique challenges posed by LLMs, including the management of massive models, the generation of quality text, and the detection of biases and errors. Specific aspects of LLMOps may include linguistic data management, language model optimization, controlled text generation and linguistic quality assessment.

An important part of our methodology is to integrate LLMOps for efficient management of the ML model lifecycle. We detail the different phases of the model lifecycle, from experimentation to production, including continuous monitoring and version management. We highlight the key features of LLMOps that facilitate task automation and the implementation of model development best practices.

### 3.3. Continuous monitoring for optimization of ML models

Continuous monitoring [32] of ML models aims to monitor in real-time the performance, behavior, and data quality of deployed models. This involves systematically collecting relevant metrics and monitoring input data to detect changes, data quality degradation, or conceptual drifts. This approach aims to ensure continuous quality and reliability of ML models by enabling real-time adjustments and improvements, which helps maintain their effectiveness and relevance in operational environments.

Emphasizing its essential role in optimizing the performance of deployed models. We discuss key performance metrics to monitor, anomaly analysis techniques and continuous improvement strategies to ensure accurate and reliable predictions. Thanks to real-time logging capabilities, performance metrics and model characteristics can be continuously monitored. This enables proactive problem detection and rapid feedback to development and operations teams. Metrics such as mean square error (MSE), coefficient of determination ( $R^2$ ) and mean absolute error (MAE) can be monitored and analyzed in real time to assess model performance.

- a. Data monitoring
  - Data statistics tracking: monitor descriptive statistics of the data, such as mean, median, standard deviation to detect significant changes in the data distribution.
  - Anomaly detection: identify outlier points in the data that could impact model performance.
  - Data quality monitoring: monitor the presence of missing values, inconsistencies, and other quality issues in the data.
- b. Model monitoring
  - Model performance tracking: monitor mean squared error (MSE), coefficient of determination ( $R^2$ ), and other evaluation metrics on the test set and production data.
  - Prediction monitoring: monitor the model predictions and identify cases where predictions are aberrant or inaccurate.
  - Model stability tracking: monitor the evolution of model coefficients and identify signs of model degradation.

### 3.4. Enabling AI explainability in artificial intelligence

Explainable artificial intelligence (XAI) [33], [34] investigates the explainability of machine learning models bridging the gap between complex model computations and human interpretability. Embedding an explainability component into the large-scale multi objective optimization problems (LMOPs) workflow brings substantial benefits that enhance the functionality and understanding of machine learning models. This addition enables a deeper grasp of the logic behind each prediction, facilitating more nuanced and informed decision-making processes. It helps in uncovering and addressing potential inaccuracies and biases within the model's outputs, thus enhancing the credibility and dependability of the results. Moreover, the provision of clear explanations allows for a better appreciation of the model's internal dynamics and the pivotal factors driving its predictions. Such transparency and insight significantly boost user trust in the model's outputs and foster greater acceptance and application of the model across various domains.

Integrating the explainability module into the MLOps pipeline allows for automatic generation of explanations for each new prediction. Explanations are stored with predictions and other metrics, enabling further analysis. The choice of explanation method and visualization depends on specific needs and user preferences. It is crucial to ensure that explanations are clear, concise, and easy to understand. The use of interactive visualization tools can also make explanations more engaging and easier to explore.

ML model explanations can be categorized into two main types: local explanations and global explanations [35], [36].

- a. Local explanations provide details about an individual prediction. Two commonly used techniques for providing local explanations are Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME) [37].
- b. SHAP: SHAP calculates the importance of each feature (area and location) for a given price prediction. It then visualizes the impact of each feature on the predicted price using SHAP bar charts.
- c. LIME: LIME generates local explanations based on simple linear models for each prediction. It identifies the most important features for a given prediction and explains their contribution.
- d. Anchor explanation: Identifies the data examples closest to a given prediction and explains why they were predicted in the same way.
- e. Global explanations aim to explain the general functioning of the model and the factors most important to its predictions. Two techniques commonly used to provide global explanations are permutation importance and partial dependence plots.
- f. Permutation importance: Permutation importance measures the importance of each feature by perturbing its order and observing the impact on model performance. This helps identify which features have the greatest impact on overall model performance.
- g. Partial dependence plots: Partial dependence plots allow you to visualize the effect of a feature on price prediction while holding other features constant. They help to understand the interaction between different characteristics and their impact on price.

The explainability module can be used in two distinct ways: as a separate component or integrated directly into the model. As a separate component, it acts as an independent tool for analyzing the predictions of an already trained model. This approach offers great flexibility, as the module can be used with different models without requiring major modifications. On the other hand, when the explainability module is integrated into the model during training, it can generate explanations directly from the model itself. This integration enables a deeper analysis of the decisions made by the model and a deeper understanding of its inner workings.

### 3.5. Fine-tuning process within LLMops

We then explore data capture and enrichment using descriptions associated with each record. We describe the process of fine-tuning [38], [39] ML models to incorporate this additional information into our dataset, in order to improve the accuracy of our prediction model. In the process of deploying a language model-based application as explained by the Figure 1, several key steps must be carefully orchestrated [40]. Firstly, selecting an appropriate base model [41] is a fundamental step. These base models, pretrained on large datasets, provide a solid foundation for a variety of subsequent tasks. Given the complexity and high cost of training such models from scratch, only a few institutions have the necessary resources to successfully undertake this challenging task. Secondly, once the base model is selected, a crucial step is to fine-tune it specifically for the envisioned downstream tasks. This fine-tuning allows customizing the model to meet the specific needs of the application in question. Once the fine-tuning is completed, it is imperative to conduct rigorous evaluation of the model to ensure its performance and reliability in real-world conditions. Finally, the last step of the process involves deploying and continuously monitoring the model in production. For this monitoring, specialized tools are emerging, such as WhyLabs or HumanLoop, allowing tracking and analyzing the model's behavior in an operational environment, thus ensuring optimal performance and proactive detection of potential degradation.

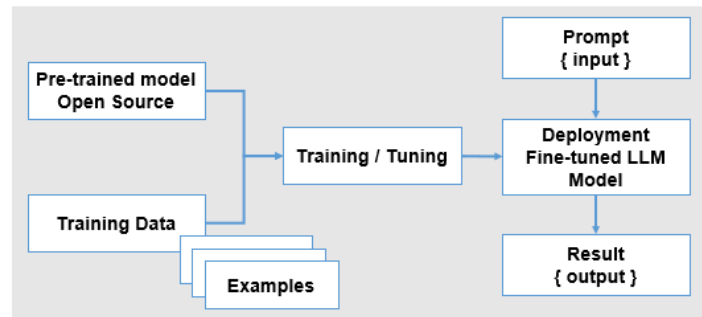


Figure 1. Fine-tuning process within the LLM framework

## 4. PROPOSED FRAMEWORK

In this section, we describe the methodology that we have adopted to build a prediction model for a specific phenomenon: dynamic optimization of ML models via MLOps and LLMops (integration of web scraping, fine-tuning, and data enrichment in a continuous monitoring context). The purpose is optimizing model prediction through a data enrichment step using LLMs and continuous monitoring.

Based on a multi-step workflow that encompasses the following stages:

- Data acquisition:** This stage involves collecting data from the target website. This can be done using techniques such as web scraping to extract relevant information from the website automatically.
- Data pre-processing:** Once the data has been collected, it needs to be cleaned and prepared for analysis. This includes the removal of outliers, management of missing data, data normalization and other pre-processing techniques to ensure the quality of the data used in the model.
- Data exploration:** This stage involves exploring and analyzing the data to understand its structure, trends and relationships. This may involve using exploratory data analysis techniques such as data visualization and statistical modeling to identify key patterns and insights.
- Model building:** Once the data has been pre-processed and explored, a prediction model is built using appropriate ML techniques such as linear regression, decision trees, and neural networks. The choice of model will depend on the characteristics of the data and the type of analysis required. The choice of model will depend on the characteristics of the data and the prediction objective.
- Model evaluation:** Once the model has been built, it is evaluated using appropriate performance measures such as mean square error (MSE) and coefficient of determination ( $R^2$ ). This allows us to determine the effectiveness of the model. This determines the model's efficiency and its ability to make accurate predictions.
- Deployment and monitoring:** Finally, once the model has been evaluated and validated, it can be deployed in a production environment to make real-time predictions. It is also important to put in place continuous monitoring mechanisms to ensure that the model remains accurate and reliable under changing conditions.

This Framework provides a structured and reproducible framework for the development and evaluation of ML models, guaranteeing the transparency and reliability of the results obtained. The diagram shown in Figure 2 represents a multi-stage workflow for the development and evaluation of a predictive ML model. In our approach, at the block level (data enrichment) the use of a LLM to exploit textual descriptions and enrich datasets can be seen as an adaptable solution for various application domains. This method makes it possible to take advantage of free descriptions in any domain to supplement dataset information. Consequently, whether in finance, healthcare, education or other sectors [42], this approach can be applied to improve the quality and diversity of available data. By integrating the advanced capabilities of language models, this method proves to be a flexible and adaptable solution to meet the specific requirements of different data analysis problems.

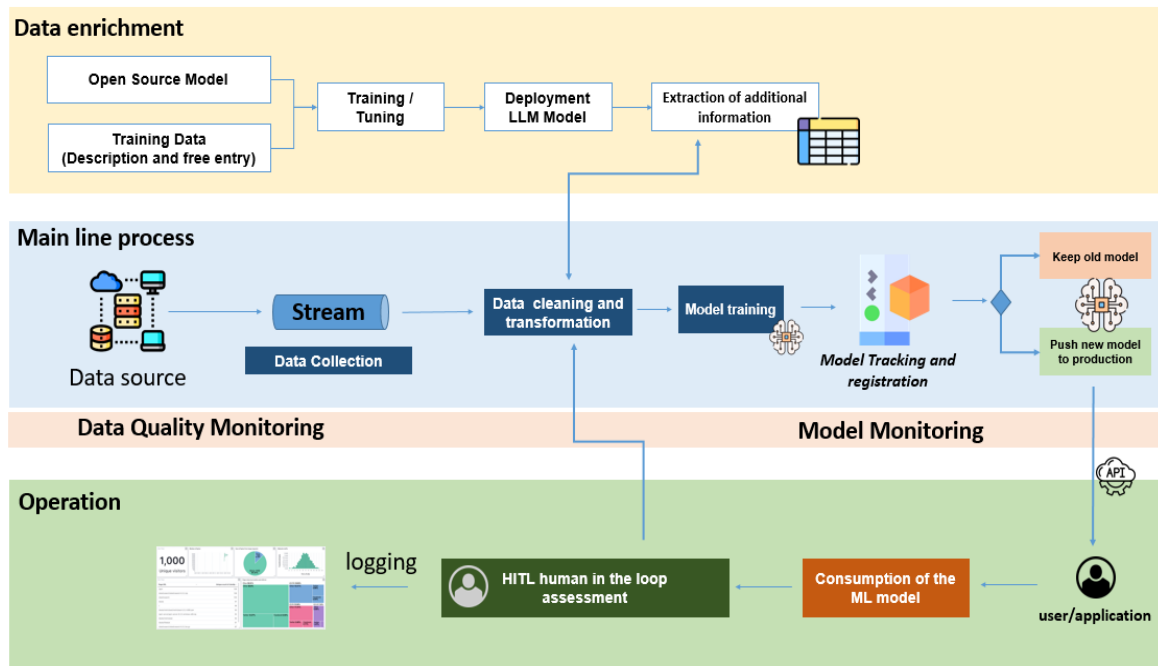


Figure 2. Optimizing model prediction: a framework integrating LLM and ML models with real-time data processing

## 5. CASE STUDY: IMPROVEMENT OF PRICE PREDICTION MODEL THROUGH LLM, LLMOPs WITH REAL-TIME DATA PROCESSING

As explained in the preceding paragraph, to enhance the predictions of the real estate price model [43], we adopted a multi-step approach detailed in the Figure 3. Firstly, we merged the information extracted by the LLMs model with the existing features, thus enriching the dataset with relevant textual data. Next, we trained a new price model using algorithms such as gradient boosting regressor or others, incorporating the enriched features. Finally, we evaluated the performance of the new price model and compared its results with those of the original model. This methodology allowed us to test the effectiveness of integrating textual data extracted by the LLMs model in enhancing the prediction performance of the real estate price model.

Integrating an MLOps framework based on MLflow into a streaming data environment represents a significant advancement in managing the lifecycle of ML models. MLflow, an open-source platform dedicated to this task, offers a multitude of functionalities that can be tailored to meet the specific requirements of streaming data. By combining MLflow with streaming data tools such as Apache Kafka and Spark Streaming [44], it becomes possible to capture and process data in real-time [45] while maintaining complete traceability of the model lifecycle. This integration not only enables real-time monitoring and management of model performance but also facilitates continuous deployment and updates of models in streaming environments. Providing a comprehensive solution for managing ML models in a streaming environment, this approach contributes to increasing the efficiency and reliability of ML systems deployed in real-time scenarios. By using MLflow to manage model deployment and updates, we can quickly deploy new versions in response to changes in data or business requirements. MLflow pipelines can be configured to

automate the process of model deployment and rollback, ensuring agile and smooth updates in a continuous streaming data environment.

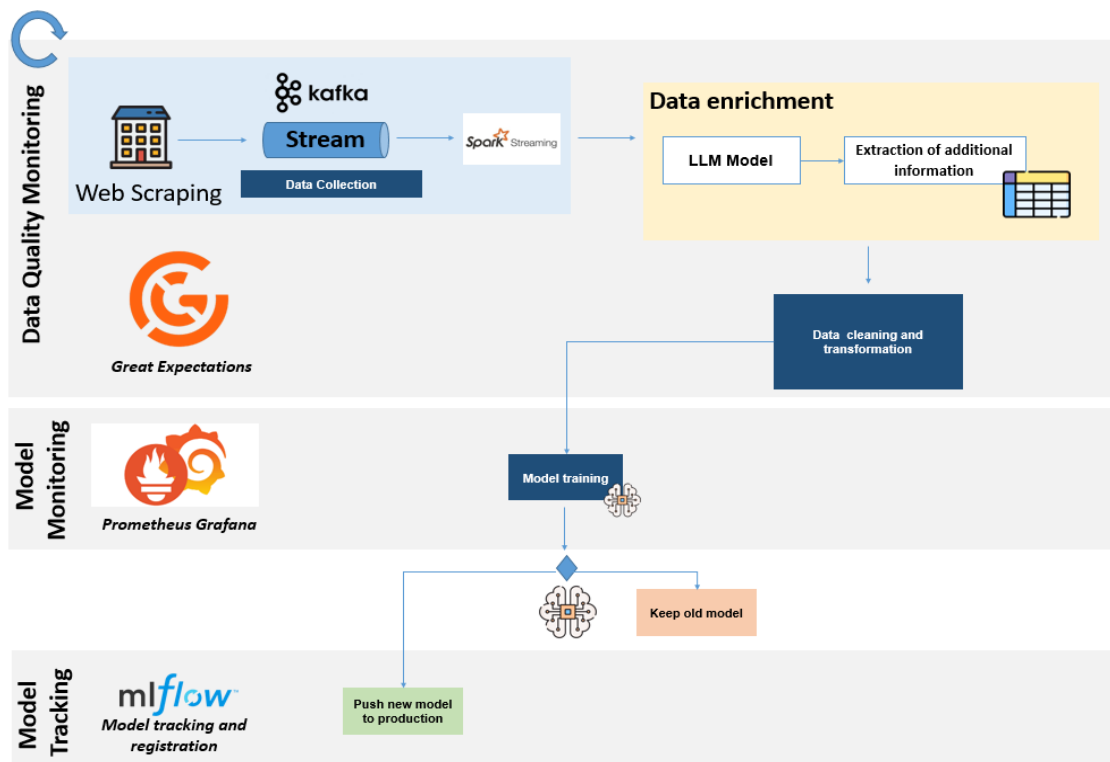


Figure 3. Improvement of price prediction model through LLM, LLMs with real-time data processing

We have placed a strong focus on data monitoring using Great Expectations and DataRobot, as well as model monitoring using Prometheus and Grafana. This comprehensive approach ensures not only the effective management of model lifecycle but also the continuous monitoring and optimization of both data and model performance in real-time scenarios. When predicting apartment real estate prices from continuously streaming data, using Great Expectations to monitor data quality can be particularly useful in ensuring reliable predictions. Indeed, Great Expectations offer the possibility of profiling data in real time, making it possible to identify and measure the essential characteristics of incoming data streams. By setting specific expectations on these data streams, such as the presence of key variables and acceptable value ranges, and regularly validating these expectations, this ensures that only high-quality data is used to drive predictive models. Furthermore, by triggering alerts in the event of deviations from these expectations, Great Expectations guarantee the reliability and consistency of the data used in property price prediction models, which is essential for accurate and reliable results in our field of application.

To illustrate the practical utility of the explainability module, let's examine two potential use scenarios in the real estate domain:

- Local explanations for real estate agents:** A real estate agent can leverage local explanations to understand the underlying reasons for a specific price prediction for a property. For example, if a property is predicted at a high price, the agent can use local explanations to identify key features that contributed to this estimate. This may include factors such as property size, location, and surrounding amenities. Such information can assist the agent in better advising clients and justifying proposed prices.
  - Global explanations for investors:** An investor seeking to acquire real estate in a specific region can use global explanations to gain insights into the most influential factors on property prices in that area. For example, by analyzing global explanations, an investor may discover that proximity to public transportation or the availability of quality schools are major determinants of prices in that region. This information can guide investment decisions by highlighting market trends and potential opportunities.
- By combining local and global explanations, stakeholders in the real estate sector can make more informed decisions, thereby maximizing their chances of success in a complex and dynamic market.

## 6. DISCUSSION

After collecting the initial data from real estate listing websites through web scraping, we encountered the reality of raw, often incomplete data. To overcome this limitation and enhance the quality of our dataset, we embarked on the subsequent phase dedicated to extracting information from the textual descriptions associated with each listing shown in Figure 4. By using the GPT API, in simple terms, we were able to extract valuable details such as specific property features, available amenities, surrounding conveniences, and more. This approach to enrichment, based on NLP, significantly improved the quality and depth of our dataset. Consequently, it paved the way for more comprehensive analyses and more relevant results in the later stages of our research. Furthermore, to keep our dataset up to date, we implemented a continuous scraping process using Kafka and Spark Streaming, ensuring that our dataset consistently reflects the evolving real estate market.

Subsequently, we opted for the use of the AutoML [46], [47] platform, a powerful tool that automates a significant portion of the machine learning model development process. Among the numerous available options such as Google AutoML, H2O.ai, Auto-sklearn, and TPOT [48], we chose auto-sklearn to address this regression problem. This platform optimizes model performance to meet our evaluation criteria. However, it is crucial to emphasize that despite the ease of use of AutoML [49], a fundamental understanding of machine learning [50] concepts remains indispensable for interpreting and fully leveraging the results produced by these tools. In our case, since stock price prediction is essentially a regression problem, we evaluate our models using metrics such as root mean squared error (RMSE), mean absolute percentage error % (MAPE), and the coefficient of determination (R2), precise measures of prediction accuracy.



Figure 4. Dataset enrichment from comments using an LLM model

The implementation of this use case within an MLOps framework has enabled us to continuously monitor the health of our captured data and the prediction quality of our machine learning model. To achieve this, we integrated several essential tools. Firstly, MLflow was used for tracking and tracing the training of our model, providing precise traceability of each iteration and its performance highlighted in Figure 5. In parallel, we established a continuous integration/continuous deployment (CI/CD) pipeline for the source code, ensuring smooth integration of updates and modifications into our production environment. Additionally, to proactively monitor the health of our system and detect potential issues, we deployed Prometheus and Grafana. These tools allow us to display and monitor essential metrics in real-time, while configuring alerts to instantly notify us of deviations or critical situations. Thus, this MLOps approach provides us with a robust framework to effectively manage our model development cycle, while ensuring the reliability and performance of our ML applications.



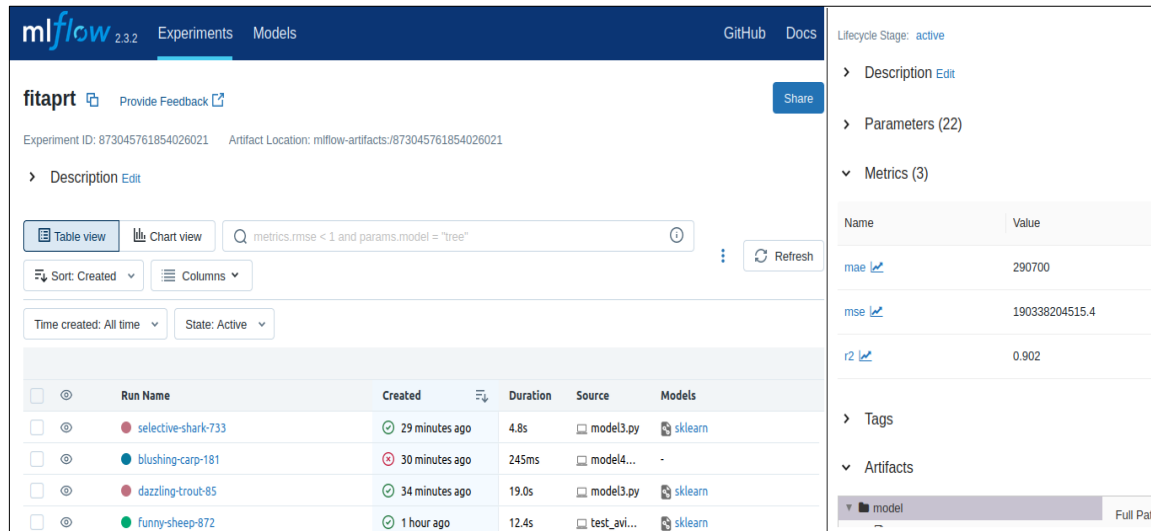


Figure 5. Experiment optimization and metric analysis with MLflow

## 7. CONCLUSION

In conclusion, the integration of MLOps into a continuous streaming data environment offers a comprehensive and agile approach for managing ML models. By monitoring model performance in real-time, quickly detecting anomalies, and enabling agile updates, MLOps allows organizations to maintain high-quality models and make informed decisions in a dynamic and evolving environment. Throughout this article, we have explored the challenges of continuous streaming data and the solutions provided by our framework, which combines MLflow and other monitoring tools for ML model management in such environments. By integrating MLOps principles and model management tools into a data streaming workflow, organizations can maximize the value of their investments in ML and maintain operational agility in a constantly changing data landscape. Additionally, defining metrics and continuous monitoring are indispensable for transitioning from a traditional exploratory environment to a high-production environment. By establishing clear metrics, organizations can better understand model performance, set benchmarks, and ensure continuous improvement. Continuous monitoring ensures that any deviations from expected performance are quickly identified and addressed, maintaining the reliability and effectiveness of ML models. Combining real-time web scraping, Kafka, Spark Streaming, and MLOps integration, our methodology offers a comprehensive approach to optimizing the real estate price prediction process. We emphasize the importance of continuous monitoring and continuous improvement to maintain high-performing ML models that are tailored to the changing requirements of the real estate market.

## REFERENCES

- [1] C. Shi, P. Liang, Y. Wu, T. Zhan, and Z. Jin, "Maximizing user experience with LLMops-driven personalized recommendation systems," *Applied and Computational Engineering*, vol. 64, no. 1, pp. 101–107, May 2024, doi: 10.54254/2755-2721/64/20241353.
- [2] A. Bodor, M. Hnida, and D. Najima, "MLOps: overview of current state and future directions," in *Innovations in Smart Cities Applications Volume 6*, Cham: Springer International Publishing, 2023, pp. 156–165.
- [3] A. Bodor, M. Hnida, and D. Najima, "From development to deployment: an approach to MLOps monitoring for machine learning model operationalization," in *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Nov. 2023, pp. 1–7, doi: 10.1109/SITA60746.2023.10373733.
- [4] E. Zimelewicz et al., "ML-enabled systems model deployment and monitoring: status quo and problems," in *Lecture Notes in Business Information Processing*, vol. 505 LNBIP, 2024, pp. 112–131.
- [5] M. Priestley, F. O'donnell, and E. Simperl, "A survey of data quality requirements that matter in ML development pipelines," *Journal of Data and Information Quality*, vol. 15, no. 2, pp. 1–39, Jun. 2023, doi: 10.1145/3592616.
- [6] S. J. Warnett and U. Zdun, "On the understandability of MLOps system architectures," *IEEE Transactions on Software Engineering*, vol. 50, no. 5, pp. 1015–1039, May 2024, doi: 10.1109/TSE.2024.3367488.
- [7] A. Kulkarni, A. Shivananda, A. Kulkarni, and D. Gudivada, "LLMs for enterprise and LLMops," in *Applied Generative AI for Beginners*, Berkeley, CA: Apress, 2023, pp. 117–154.
- [8] A. Kulkarni, A. Shivananda, A. Kulkarni, and D. Gudivada, "Implement LLMs using Sklearn," in *Applied Generative AI for Beginners*, Berkeley, CA: Apress, 2023, pp. 101–116.
- [9] V. Kozov, G. Ivanova, and D. Atanasova, "Practical application of AI and large language models in software engineering education," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 1, 2024, doi: 10.14569/IJACSA.2024.0150168.
- [10] G. G. Krishna, "Multilingual NLP," *International Journal of Advanced Engineering and Nano Technology*, vol. 10, no. 6, pp. 9–12, Jun. 2023, doi: 10.35940/ijaent.E4119.0610623.




*Integration of web scraping, fine-tuning, and data enrichment in a continuous monitoring ... (Anas Bodor)*

- [11] S. Kamath Barkur, P. Sitapara, S. Leuschner, and S. Schacht, "Magenta: metrics and evaluation framework for generative agents based on LLMs," *Intelligent Human Systems Integration (IHSI 2024): Integrating People and Intelligent Systems*, 2024, doi: 10.54941/ahfe1004478.
- [12] L. Budach *et al.*, "The effects of data quality on ML-model performance," *CoRR abs/2207.14529*, 2022.
- [13] R. Marpu and B. Manjula, "Streaming machine learning algorithms with streaming big data systems," *Brazilian Journal of Development*, vol. 10, no. 1, pp. 322–339, Jan. 2024, doi: 10.34117/bjdv10n1-021.
- [14] M. Y. E. Saputra, Noprianto, S. Noor Arief, V. N. Wijayaningrum, and Y. W. Syaifudin, "Real-time server monitoring and notification system with prometheus, Grafana, and Telegram integration," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, Jan. 2024, pp. 1808–1813, doi: 10.1109/ICETSIS61505.2024.10459488.
- [15] S. C. Huang and T. H. Le, *Principles and labs for deep learning*. United Kingdom: Academic Press, 2021.
- [16] A. Bodor, M. Hnida, and N. Daoudi, "Machine learning models monitoring in MLOps context: metrics and tools," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 17, no. 23, pp. 125–139, Dec. 2023, doi: 10.3991/ijim.v17i23.43479.
- [17] "Great expectations," <https://greatexpectations.io/> (accessed Aug. 01, 2024).
- [18] J. Krzywanski *et al.*, "AutoML-based predictive framework for predictive analysis in adsorption cooling and desalination systems," *Energy Science & Engineering*, vol. 12, no. 5, pp. 1969–1986, May 2024, doi: 10.1002/ese3.1725.
- [19] D. Petrova-Antonova and R. Tancheva, "Data cleaning: a case study with openrefine and trifacta wrangler," in *Communications in Computer and Information Science*, vol. 1266 CCIS, 2020, pp. 32–40.
- [20] A. Pandey, M. Sonawane, and S. Mamtani, "Deployment of ML models using kubeflow on different cloud providers," *arXiv preprint arXiv:2206.13655*, 2022.
- [21] L. Berberi, V. Kozlov, K. Alibabaei, and B. Esteban, "MLflow and its usage," *arXiv preprint arXiv*, 2022.
- [22] "Seldon," <https://www.seldon.io/> (accessed Aug. 01, 2024).
- [23] A. H. Ali, M. Alajanbi, M. G. Yaseen, and S. A. Abed, "Chatgpt4, DALL·E, Bard, Claude, BERT: open possibilities," *Babylonian Journal of Machine Learning*, vol. 2023, pp. 17–18, Mar. 2023, doi: 10.58496/BJML/2023/003.
- [24] F. Alhaj, A. Al-Haj, A. Sharieh, and R. Jabri, "Improving arabic cognitive distortion classification in twitter using BERTopic," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, 2022, doi: 10.14569/IJACSA.2022.0130199.
- [25] E. Yulianti, N. Pangestu, and M. A. Jiwanggi, "Enhanced textrank using weighted word embedding for text summarization," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5472–5482, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5472-5482.
- [26] D. Dillion, N. Tandon, Y. Gu, and K. Gray, "Can AI language models replace human participants?," *Trends in Cognitive Sciences*, vol. 27, no. 7, pp. 597–600, Jul. 2023, doi: 10.1016/j.tics.2023.04.008.
- [27] Z. He *et al.*, "Exploring human-like translation strategy with large language models," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 229–246, Mar. 2024, doi: 10.1162/tacl\_a\_00642.
- [28] I. O. William and M. Altamimi, "Large language model for creative writing and article generation," *International Journal of Advanced Natural Sciences and Engineering Researches*, pp. 741–748, 2024.
- [29] D. Huang *et al.*, "DSQA-LLM: domain-specific intelligent question answering based on large language model," in *Communications in Computer and Information Science*, 2024, vol. 1946 CCIS, pp. 170–180, doi: 10.1007/978-981-99-7587-7\_14.
- [30] M. Skorikov, K. N. J. Omar, and R. Khan, "Voice-controlled intelligent personal assistant," in *Smart Innovation, Systems and Technologies*, vol. 273, 2022, pp. 57–65.
- [31] F. Antonius *et al.*, "Incorporating natural language processing into virtual assistants: an intelligent assessment strategy for enhancing language comprehension," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, 2023, doi: 10.14569/IJACSA.2023.0141079.
- [32] R. Bagai, A. Masrani, P. Ranjan, and M. Najana, "Implementing continuous integration and deployment (CI/CD) for machine learning models on AWS," *International Journal of Global Innovations and Solutions (IJGIS)*, May 2024, doi: 10.21428/e90189c8.9cb39c55.
- [33] T.-C. T. Chen, "Explainable artificial intelligence (XAI) with applications," in *Explainable Ambient Intelligence (XAmI) Explainable Artificial Intelligence Applications in Smart Life*, Springer, 2024, pp. 23–38.
- [34] S. Roy, G. Laberge, B. Roy, F. Khomh, A. Nikanjam, and S. Mondal, "Why don't XAI techniques agree? characterizing the disagreements between post-hoc explanations of defect predictions," in *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Oct. 2022, pp. 444–448, doi: 10.1109/ICSME55016.2022.00056.
- [35] X. Kong, S. Liu, and L. Zhu, "Toward human-centered XAI in practice: a survey," *Machine Intelligence Research*, vol. 21, no. 4, pp. 740–770, Aug. 2024.
- [36] S. Alam and Z. Altıparmak, "XAI-CF-examining the role of explainable artificial intelligence in cyber forensics," *arXiv preprint arXiv:2402.02452*, 2024.
- [37] L. Schulte, B. Ledel, and S. Herbold, "Studying the explanations for the automated prediction of bug and non-bug issues using LIME and SHAP," *Empirical Software Engineering*, vol. 29, no. 4, Jul. 2024, doi: 10.1007/s10664-024-10469-1.
- [38] A. López-López, J. M. Garcia-Gorrostieta, and S. González-López, "Emotion detection in educational dialogues by transfer learning," *Journal of Intelligent & Fuzzy Systems*, pp. 1–11, Mar. 2024, doi: 10.3233/JIFS-219340.
- [39] X. Li, Y. Zhang, and E. C. Malthouse, "Exploring fine-tuning ChatGPT for news recommendation," *arXiv preprint arXiv:2311.05850*, 2023.
- [40] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Next-generation spam filtering: comparative fine-tuning of LLMs, NLPs, and CNN models for email spam classification," *Electronics*, vol. 13, no. 11, May 2024, doi: 10.3390/electronics13112034.
- [41] M. Seiranian, "Large language model parameter efficient fine-tuning for mathematical problem solving final project report," 2024, doi: 10.13140/RG.2.2.28262.84806.
- [42] A. Sharma, K. K. Upman, D. Saini, and A. Raj, "NLP and it's all application in AI," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 43, no. 4, pp. 180–183, Nov. 2023, doi: 10.52783/tjjpt.v43.i4.2328.
- [43] S. Abdul-Rahman, N. H. Zulkifley, I. Ibrahim, and S. Mutalib, "Advanced machine learning algorithms for house price prediction: case study in Kuala Lumpur," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, 2021, doi: 10.14569/IJACSA.2021.0121291.
- [44] D. S. Mphasis and D. Seenivasan, "Real-time data processing with streaming ETL," *Article in International Journal of Science and Research*, vol. 12, pp. 2185–2192, 2023, doi: 10.21275/SR24619000026.
- [45] N. B. Kilaru, "Design real-time data processing systems for ai applications," *International Journal for Research Publication and Seminar*, vol. 15, no. 3, pp. 472–481, Sep. 2024, doi: 10.36676/jrps.v15.i3.1538.




- [46] J. Wu, H. Wang, C. Ni, C. Zhang, and W. Lu, "Data pipeline training: integrating autoML to optimize the data flow of machine learning models," in *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, Mar. 2024, pp. 730–734, doi: 10.1109/ICAACE61206.2024.10549260.
- [47] M. A. Al Alamin and G. Uddin, "How far are we with automated machine learning? characterization and challenges of AutoML toolkits," *Empirical Software Engineering*, vol. 29, no. 4, Jul. 2024, doi: 10.1007/s10664-024-10450-y.
- [48] F. Stoica and L. Florentina Stoica, "AutoML insights: gaining confidence to operationalize predictive models," in *The New Era of Business Intelligence [Working Title]*, IntechOpen, 2024.
- [49] R. Cheng, H. J. Escalante, W.-W. Tu, J. N. Van Rijn, S. Wang, and Y. Yang, "Guest editorial: autoML for nonstationary data," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 2456–2457, Jun. 2024, doi: 10.1109/TAI.2024.3387583.
- [50] I. Slimani *et al.*, "Automated machine learning: the new data science challenge," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, pp. 4243–4252, Aug. 2022, doi: 10.11591/ijece.v12i4.pp4243-4252.

## BIOGRAPHIES OF AUTHORS






**Anas Bodor**    head of the quality, standards, and technological monitoring unit at the Moroccan Statistical office. He is also a Ph.D. student at the LyRica Lab in the School of Information Sciences, Rabat, Morocco. He can be contacted at email: [anas.bodor@esi.ac.ma](mailto:anas.bodor@esi.ac.ma).



**Meriem Hnida**    assistant professor at the information sciences school, PhD in computer sciences at the Mohammadia School of Engineering (2019). Permanent member of the ITQAN research team, LYRICA laboratory, and associate member of "networking, modeling and e-learning (RIME)" research team of Mohammadia School of Engineering. Her research project is about the application of intelligent systems in education, and focuses on the following research areas: intelligent tutorial systems (ITS), educational technology, knowledge engineering. She can be contacted at email: [mhnida@esi.ac.ma](mailto:mhnida@esi.ac.ma).



**Najima Daoudi**    full professor at the School of Information Sciences, Rabat, Morocco. She has an engineering degree from the National Institute of Statistics and Applied Economics (INSEA) and a Ph.D. in computer science from ENSIAS. She can be contacted at email: [ndaoudi@esi.ac.ma](mailto:ndaoudi@esi.ac.ma).