Vol. 16, No. 4, 2025
ISSN 2233-4203/ e-ISSN 2093-8950
ARTICLE www.msletters.org | Mass Spectrometry Letters

A Pilot Study on Using Large Language Models to Simplify Machine Learning for Mass Spectrometry-Based Classification of Erectile Dysfunction Drug Analogues

Eunwoo Choi and Han Bin Oh*

Department of Chemistry, Sogang University, Seoul 04107, Republic of Korea

Received September 30, 2025, Revised December 12, 2025, Accepted December 17, 2025
First published on the web December 31, 2025; DOI: 10.5478/MSL.2025.16.4.111

Abstract: The recent emergence of large language models (LLMs) has transformed the process of machine learning (ML) model development, markedly reducing the need for advanced coding expertise and enabling domain scientists to directly construct computational workflows through natural language prompts. In this study, we demonstrate the application of Google Gemini Pro 2.5 for developing classification models of erectile dysfunction (ED) drug analogues using tandem mass spectrometric (MS/MS) data. The dataset consisted of 149 compounds, including sildenafil, vardenafil, tadalafil analogues, and structurally unrelated compounds, represented as binary barcode spectra (*m/z* 50–800) derived from fragment ion intensities. Through stepwise prompting, the LLM generated executable Python code for data preprocessing, model construction, hyperparameter optimization, and ensemble learning using random forest, artificial neural networks (ANN), and support vector machines (SVM). The resulting models achieved high classification performance comparable to that of a manually programmed ANN reported in our previous work, while requiring markedly less programming effort. Beyond reproducing classification accuracy, this study highlights the efficiency, accessibility, and reproducibility of LLM-assisted ML workflows, underscoring their potential to democratize computational methods in mass spectrometry and analytical chemistry.

Keywords: machine learning, large language model (LLM), Erectile dysfunction drugs, mass spectrometry, forensic analysis

Introduction

In recent years, the environment for developing machine learning (ML) models has undergone a significant transformation. Traditionally, building advanced ML models requires an in-depth knowledge of specific programming languages such as Python or MATLAB, along with substantial coding effort and debugging skills.[1–5] These requirements created a high barrier to entry for researchers in specialized fields like analytical chemistry, who often possess deep domain expertise but may have limited programming experience. However, the emergence of large language models (LLMs), particularly state-of-the-art generative AI systems such as OpenAI's ChatGPT and Google's Gemini, is fundamentally changing this paradigm.[6-8]

LLMs exhibit an extraordinary capacity to understand natural language prompts and translate them into executable code, holding the potential to streamline and accelerate the ML model development process.[8] Through interactive, iterative prompting, users can now generate, refine, and implement ML pipelines without extensive prior knowledge of programming or ML theory. This capability opens new avenues for applying ML across various scientific disciplines, including analytical chemistry.[9,10]

Within this context of technological transformation, the present study aims to empirically demonstrate how readily ML models can be developed using LLMs to address domain-specific challenges in analytical chemistry. As a case study, we revisit and extend our previously published research from 2019[1], in which an artificial neural network (ANN) was developed to classify unknown erectile dysfunction (ED) drugs and their analogues into four categories, i.e., sildenafil, vardenafil, tadalafil and others, based on tandem mass spectral data acquired via liquid chromatography–tandem mass spectrometry (LC–MS/MS).

LC–MS/MS remains a powerful and widely utilized analytical technique, especially suited for the identification and quantification of small-molecule drugs and their analogues in complex biological or food matrices.[11,12] Its high sensi-

*Reprint requests to Han Bin Oh
https://orcid.org/0000-0001-7919-0393
E-mail: hanbinoh@sogang.ac.kr

tivity and specificity make it indispensable in both regulatory and forensic contexts. In the case of ED drugs, this capability is particularly critical: unapproved analogues are frequently and illicitly added to over-the-counter dietary supplements, posing significant risks to public health.[13] These illicit formulations often contain undeclared active pharmaceutical ingredients (APIs) or uncharacterized structural analogues with unknown toxicological profiles[14], which can lead to severe and unpredictable adverse effects in consumers. Accordingly, accurate classification and detection of these analogues is not only a scientific challenge but also a regulatory necessity.

The ANN model in the 2019 study was implemented directly in MATLAB–then a common choice for customized ML development in the scientific community.[1] While the study demonstrated the feasibility of applying ML techniques to LC–MS/MS data for drug classification, the process required considerable programming expertise, time, and iterative debugging. These barriers limited the accessibility of such methods to a broader range of analytical researchers.

The primary objective of this study is to reproduce the analytical task addressed in the 2019 research–developing a classification model for ED drug analogues based on LC–MS/MS data–using an LLM, specifically Google Gemini Pro 2.5. Rather than focusing solely on achieving equivalent classification performance, this study emphasizes the efficiency, accessibility, and reproducibility of the model development process when assisted by LLMs.

Modern LLMs such as ChatGPT and Gemini possess robust reasoning capabilities and support for multiple programming languages, including Python.[15] They can generate and refine code for data preprocessing, model construction (e.g., ANN), training, and evaluation, all through human-readable natural language instructions.[15] The resulting code typically incorporates widely used libraries such as pandas, scikit-learn, and TensorFlow/Keras, making it easy to interpret, extend, and deploy.

This LLM-assisted approach has the potential to democratize machine learning by enabling researchers to focus on problem formulation, result interpretation, and scientific inquiry, rather than technical implementation details. For analytical chemists with limited programming backgrounds, this represents a transformative opportunity to adopt ML methods more readily and effectively. Through the concrete example of building a classification model for ED drug analogues, this study illustrates how LLMs can lower the barrier to entry and accelerate the development of domain-specific computational tools. Beyond mere efficiency in code generation, this paradigm shift opens new possibilities for advancing analytical problem-solving and accelerating scientific discovery.

Computations

Data

The tandem mass spectral data used in this study are identical to those employed in our previous work.[1] Briefly, the dataset comprises tandem mass spectra for 34 sildenafil analogues, 8 vardenafil analogues, 14 tadalafil analogues, and 131 additional compounds classified as 'others'. The chemical names, formulas, molecular weights, SMILES notations, and structures of these compounds are provided in Supporting Information. All spectra were acquired in positive-ion mode using an Agilent 6530 hybrid quadrupole time-of-flight (Q-TOF) mass spectrometer (Agilent Technologies, Palo Alto, CA, USA). A complete list of the compounds is provided in Supporting Information.

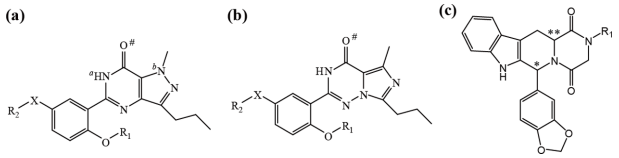

Figure 1. Three major classes of erectile dysfunction (ED) drugs: (a) sildenafil, (b) vardenafil, and (c) tadalafil.

Figure 1 illustrates the representative chemical structures of the three major classes of erectile dysfunction (ED) drugs: sildenafil, vardenafil, and tadalafil. These compounds are phosphodiesterase-5 (PDE5) inhibitors. As shown in this figure, ED drugs are broadly classified into these three structurally distinct categories. Although sildenafil and vardenafil share general pharmacological features as PDE5 inhibitors, they differ in their core heterocyclic frameworks: sildenafil features a pyrazolopyrimidinone scaffold, while vardenafil is based on an imidazo[2,1-f]pyridazinone core. In contrast, tadalafil displays a unique β-carboline-based structure, making it markedly different from the other two classes.

LLM and computational resources

Google Gemini 2.5 Pro (version as of June 12, 2025) was used as a representative example of a large language model (LLM). The Python code generated by the LLM was executed in the Google Colab environment. All code was run using Python 3.11 on a Google Colab instance configured with the e2-standard-4 machine type, without the use of any hardware accelerators.

Access to Google Gemini 2.5 Pro and the Google Colab environment was made from a personal computer equipped with an AMD Ryzen 9 7950X 16-Core Processor (4.50 GHz) and 64 GB of RAM), running Windows 11 Home (version 24H2).

Results and discussion

Data Preparation for LLM-Assisted Machine Learning Modelling

In this study, we utilized the same dataset that was used in our previous 2019 publication to ensure direct comparabil-

Table 1. Example of input data in barcode format. Labels 1, 2, 3, and 4 correspond to the ED classes of sildenafil, vardenafil, tadalafil, and others, respectively.

Index	Name	*m/z*	Label	50	51	…	297	298	299	300	301	302	…	799	800
1	Sildenafil	475.2121	1	0	0	…	1	0	1	0	0	0	…	0	0
2	Vardenafil	489.2278	2	0	0	…	0	0	1	0	0	0	…	0	0
3	Tadalafil	390.1448	3	0	0	…	0	0	0	0	0	1	…	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
187	Velutin	315.0863	4	0	0	…	0	0	0	1	1	0	…	0	0

ity.[1] Specifically, tandem mass spectra of a total of 149 erectile dysfunction (ED) drug analogues and other compounds were processed using a binning approach across the *m/z* range of 50 to 800, with a bin size of 1 *m/z* unit. For each spectrum, a binary barcode was generated based on fragment ion intensities: peaks exceeding a defined intensity threshold, e.g., S/N ratio ≥3 in this study, were assigned a value of "1", while those below the threshold were assigned a value of "0 (or null)".

The rationale for adopting this binary encoding, rather than using raw or normalized intensity values, lies in the inherent variability of intensity profiles in tandem mass spectra. These variations are largely influenced by instrument-specific parameters, especially collision energy. Accounting for all such factors would dramatically increase the number of spectra required and introduce significant weighting from arbitrary experimental conditions. Thus, the binary barcode approach was chosen to simplify the data while minimizing the impact of instrument-dependent variability. However, this simplification comes with a trade-off: the loss of relative intensity information between fragment ions may reduce the classification performance of machine learning models, as some structural features are encoded in the intensity patterns of the spectra.

Table 1 presents an example of the input dataset of representative compounds prepared for machine learning modelling, saved in CSV format. Each row contains the compound name, its *m/z* value, classification label (e.g., sildenafil, vardenafil, tadalafil, or others), and the corresponding binary barcode spectrum based on the presence or absence of peaks at each *m/z* value. This dataset was then used for LLM-assisted machine learning model development. For modelling, the classification labels (e.g., sildenafil, vardenafil, tadalafil, or others) were converted to numerical codes (sildenafil = 1, vardenafil = 2, tadalafil = 3, others = 4) for the input file.

LLM prompts

The following instructions were provided to the LLM stepwise rather than as a single prompt. Although the original instructions were given in Korean, they have been translated into English for inclusion in this manuscript. The version of Google Gemini 2.5 Pro used in this study demonstrated strong compatibility with the Korean language, successfully interpreting and executing the prompts even when they did not strictly adhere to standard grammatical structures.

A systematic strategy is essential when developing machine learning models, and it is particularly important to outline the overall plan in the first prompt. The following set of prompts is intended for situations in which the user has limited knowledge of machine learning and is unsure of the specific parameters required for effective model building. In such cases, it can be helpful first to provide a general plan and then request assistance in refining the prompts by incorporating appropriate parameter specifications.

1st Prompt:

"The following is the first prompt for a new project. Additional prompts–specifically the second and third–will be provided sequentially. Please review this initial prompt and await the subsequent ones. After reviewing the second and third prompts, kindly carry out the task described below.

The objective of this project is to generate Python code for machine learning model development using a large language model (LLM). In the second prompt, I will describe the structure and content of the input dataset. In the third prompt, I will provide instructions for implementing specific machine learning models. However, due to my limited knowledge of machine learning, I am not familiar with the detailed hyperparameters required to build optimal models. Therefore, I would greatly appreciate it if you could review both the second and third prompts and help refine the third prompt by suggesting more specific and appropriate hyperparameter settings.

After that, I will provide the refined version of the third prompt. Once it is available, please use it to generate Python code for building machine learning models based on the second input dataset."

2nd Prompt:

"This file (path of data file. ex.'/content/drive/MyDrive/SIDA_transfer/input_re.csv') is a dataset intended for machine learning. It contains digital information derived from the tandem mass spectra of various compounds. Our primary interest lies in erectile dysfunction (ED) drug analogues, which include sildenafil, vardenafil, and tadalafil.

In addition to these compounds, the dataset also contains tandem mass spectral data for other substances not classified as ED drug analogues, labelled as 'others'. In this CSV file, compounds corresponding to Class 1 are sildenafil analogues; Class 2, vardenafil analogues; Class 3, tadalafil analogues; and Class 4, other non-ED compounds. The class information is found in the fourth column (see Table 1). The feature matrix begins from the fifth column and extends to the final column (751 data columns in total). Data entries start from the second row and continue to the end of the file (187 data rows in total)."

In this step, the LLM was provided with a detailed explanation of the input data file, as shown in Table 1. This included descriptions of classification labels, the location of class column data, and the starting row for compound entries.

3rd Prompt:

"Using the tandem mass spectra contained in the CSV file described above as input, please generate Python code snippets for optimal machine learning classification modelling. The full dataset should be split into a training dataset and an external test dataset at a ratio of 8:2. This split should maintain the class distribution (i.e., stratified sampling). The model validation should be performed using 5-fold cross-validation. Please implement three different models: random forest, artificial neural network (ANN), and support vector machine (SVM). When performing GridsearchCV, use the widest reasonable range of hyperparameters to ensure comprehensive exploration of the model's performance space. Furthermore, the code should include an ensemble prediction step that performs hard voting across the three models. In cases where all three models yield different predictions, the code should output all individual model predictions for reference. The results for both the training and external test datasets should be reported in the form of classification reports and confusion matrices. For the confusion matrix visualization, use black-and-white formatting, ensuring the parts on the diagonal are displayed in light gray. Arrange the matrices in a 2-by-2 subplot layout. The classification report should be exported as a text file. The code should also include functionality to export the trained models for future use. Finally, the code must calculate and print the total execution time."

Based on this, the LLM generates a revised third prompt. The newly generated third prompt is available in Supporting Information. Below, we provide representative examples of the hyperparameters applied in each of the three machine learning models. We entered this prompt but the prompt is too long. We divided the prompt into two parts and entered them one by one.

- Random Forest Classifier:
  - n_estimators: [100, 200, 300, 500] (Number of trees in the forest)
  - max_features: ['sqrt', 'log2', 0.6, 0.8] (Number of features to consider when looking for the best split)
  - max_depth: [10, 20, 30, None] (Maximum depth of the tree; None means unlimited)
  - min_samples_split: [2, 5, 10] (Minimum number of samples required to split an internal node)
  - min_samples_leaf: [1, 2, 4] (Minimum number of samples required to be at a leaf node)
  - bootstrap: [True, False] (Whether bootstrap samples are used when building trees)
- Artificial Neural Network (ANN) - MLPClassifier:
  - hidden_layer_sizes: [(50,), (100,), (50, 50), (100, 50), (100, 100)] (Tuple representing the number of neurons in each hidden layer)
  - activation: ['relu', 'tanh'] (Activation function for the hidden layer)
  - solver: ['adam', 'sgd'] (Solver for weight optimization)
  - alpha: [0.0001, 0.001, 0.01] (L2 penalty (regularization term) parameter)
  - learning_rate: ['constant', 'adaptive'] (Learning rate schedule for weight updates)
  - max_iter: [500, 1000, 1500] (Maximum number of iterations)
- Support Vector Machine (SVM) - SVC:
  - C: [0.1, 1, 10, 100] (Regularization parameter. The strength of the regularization is inversely proportional to C.)
  - kernel: ['linear', 'rbf', 'poly'] (Specifies the kernel type to be used in the algorithm)
  - gamma: ['scale', 'auto', 0.001, 0.01, 0.1] (Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. 'scale' uses 1 / (n_features * X.var()), 'auto' uses 1 / n_features.)
  - degree: [2, 3, 4] (Degree of the polynomial kernel function ('poly'). Ignored by other kernels.) (This will only be relevant for kernel='poly')

In the refined third prompt, the LLM was provided with detailed instructions for developing machine learning models based on the input data described in the first prompt. The task focused on constructing Random Forest, Artificial Neural Network (ANN), and Support Vector Machine (SVM) classifiers, and integrating their outputs through a majority voting strategy (hard voting) to generate final predictions. This ensemble approach was employed to mitigate the individual limitations of each model while capitalizing on their complementary strengths, thereby enhancing the overall robustness and reliability of the results.

It is worth noting that the above prompts represent just one possible approach to achieving the intended modelling outcome; alternative prompt configurations may also be employed. Furthermore, users can append new prompts or modify existing ones as needed to refine or extend the modelling process.

LLM-generated Python code and modelling results

Although the LLM is capable of generating machine learning (ML) code and workflows, direct execution within

the LLM interface frequently resulted in implementation errors. These issues were likely due to limitations in the LLM's interactive execution capabilities or ambiguities in prompt interpretation, rather than insufficient computational resources. To address this, we used the LLM solely for code generation, with execution carried out in the Google Colab environment. To ensure reproducibility during data splitting, each modelling run was configured with the same random state parameter–specifically, 42–while maintaining class distribution through stratified sampling. The code was designed to require manual renaming of output files before each execution, allowing for iterative modelling and the organized storage of results from each run. Although manual renaming was used in this study, the code can be easily modified to assign unique file names dynamically, enabling further automation if desired. Alternatively, this can be addressed through separate prompting.

On the system used for this study, each modelling iteration required approximately 0.47 seconds, and the full optimization process, including multiple iterations, was completed within a few hours.

Users should ensure that the input file path within the code is adjusted to reflect the actual storage location of their dataset. The results obtained from executing this code in Google Colab are summarized in Figure 2, which presents the confusion matrices and classification reports for the final ensemble model selected via a hard voting.

It is notable that the hard-voting ensemble yielded superior performance metrics compared to the individual models–Random Forest, ANN, and SVM–across multiple evaluation criteria. Given the relatively small size of the dataset, these outcomes are considered highly satisfactory. Other details of statistical metrics like precision are shown in Table 2. The detailed definitions of the mentioned metrics related to confusion matrix are given in the Supporting Information.

For the entire dataset, the ensemble model achieved an accuracy of 99%, sensitivity (recall) of 100%, except for Vardenafil (75%), and specificity of 100% at vardenafil and tadalafil, 99% at sildenafil, 98% at others. In the case of vardenafil, the relatively low recall value is likely attributable to the limited number of available samples and its high structural similarity to sildenafil, which together suggest that the model was not sufficiently trained to distinguish vardenafil with high confidence. Nevertheless, this limitation is expected to diminish as more tandem mass spectral data become available, enabling more robust and comprehensive model development in future studies.

Compared with the previous work in 2019, which implemented an ANN model in MATLAB, the current study employs LLM-assisted Python-based modeling and integrates multiple classifiers, including Random Forest, ANN, and SVM, combined through a hard-voting ensemble. While the earlier work reported 100% accuracy in its evaluation, the ensemble model in the present study achieves an overall accuracy of 99%, with a notably lower recall of 0.75 for vardenafil.

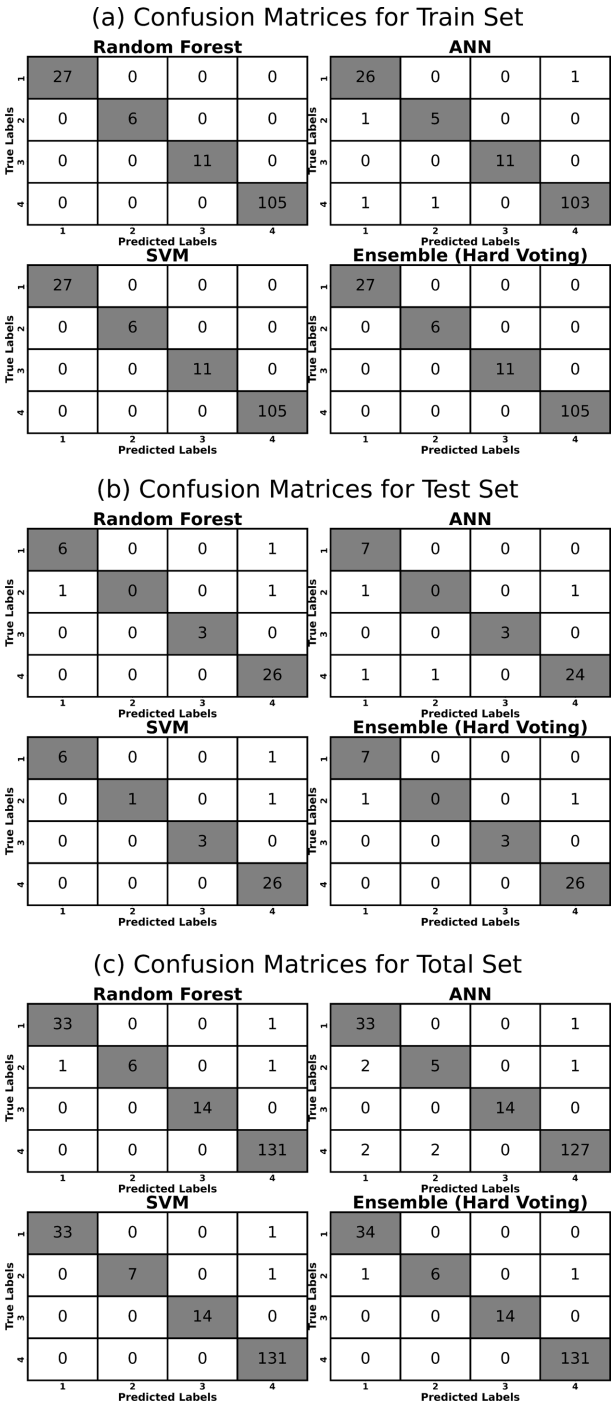

Figure 2. Confusion matrices of hard voting ensemble approach: (a) the train set, (b) the test set, and (c) the total set.

Coping with error messages

Most error messages occurred when the input file path

Table 2. Statistical reports of the classification models: top, Random Forest; middle upper, ANN; middle lower, SVM; bottom, the hard-voting ensemble. This dataset includes results from both the training set and the test set.

Class	Precision	Recall	F1-Score
Sildenafil (1)	0.97	0.97	0.97
Vardenafil (2)	1.00	0.75	0.86
Tadalafil (3)	1.00	1.00	1.00
Others (4)	0.98	1.00	0.99
Accuracy			0.98
Macro avg	0.99	0.93	0.96
Weighted avg	0.98	0.98	0.98

Class	Precision	Recall	F1-Score
Sildenafil (1)	0.89	0.97	0.93
Vardenafil (2)	0.71	0.62	0.67
Tadalafil (3)	1.00	1.00	1.00
Others (4)	0.98	0.97	0.98
Accuracy			0.96
Macro avg	0.90	0.89	0.89
Weighted avg	0.96	0.96	0.96

Class	Precision	Recall	F1-Score
Sildenafil (1)	1.00	0.97	0.99
Vardenafil (2)	1.00	0.88	0.93
Tadalafil (3)	1.00	1.00	1.00
Others (4)	0.98	1.00	0.99
Accuracy			0.99
Macro avg	1.00	0.96	0.98
Weighted avg	0.99	0.99	0.99

Class	Precision	Recall	F1-Score
Sildenafil (1)	0.97	1.00	0.99
Vardenafil (2)	1.00	0.75	0.86
Tadalafil (3)	1.00	1.00	1.00
Others (4)	0.99	1.00	1.00
Accuracy			0.99
Macro avg	0.99	0.94	0.96
Weighted avg	0.99	0.99	0.99

was incorrectly specified, requiring verification of the file path. Additional errors occasionally arose during code execution. A particularly useful feature in Google Colab is the appearance of a "Next step: Explain error" button below error messages. Clicking this button launches a new Gemini instance on the right side of the screen, which provides a clear explanation of the error along with suggested solutions.

For instance, in response to the error message

"AttributeError: 'numpy.ndarray' object has no attribute 'map'",

Google Colab recommended:

"Convert numpy arrays to pandas Series before mapping."

This clarified that the issue was related to data types, and converting the ndarray to a Series successfully resolved the problem. Additionally, Colab provided the corrected code, allowing the user to apply the fix directly with minimal effort.

Additional works

Each mass spectrometer manufacturer typically uses a proprietary data format for storing mass spectrometry data; however, these files can be readily converted to standardized formats such as mzXML or mzML using freely available tools. For example, the MSConvert program (ProteoWizard, http://proteowizard.sourceforge.net/download.html) allows easy conversion to the mzXML format. Furthermore, these converted data files can also be processed using LLMs to transform them into a format compatible with the machine learning models developed in this study. The LLM prompt used for this transformation is presented in the Supporting Information.

Conclusions

This study demonstrates that large language models (LLMs) can substantially reduce the technical barriers to applying machine learning in analytical chemistry. By employing Google Gemini Pro 2.5, we reproduced and extended a classification workflow for erectile dysfunction drug analogues using tandem mass spectrometric data, demonstrating results comparable to those obtained with conventional, manually coded approaches but with far greater efficiency and accessibility. Importantly, while Gemini Pro 2.5 was utilized in this work, comparable performance is expected from other advanced LLMs with similar capabilities. Given the rapid pace of progress in generative AI, future LLMs will likely offer even more powerful, accurate, and versatile functionalities, further enhancing the reproducibility and democratization of machine learning in mass spectrometry and related fields. These findings underscore the transformative potential of LLMs to accelerate computational method development and broaden participation in data-driven chemical research.

Acknowledgments

This work is supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) [2021R1A2C2007397] and by the Ministry of Education [2018R1A6A1A03024940] through Basic Science Research Program. OHB is thankful to the grant

(RS-2024-00332095) from Ministry of Food and Drug Safety (MFDS) and the Korea Health Industry Development Institute (KHIDI), and also to Korea Environment Industry & Technology Institute (KEITI) through "Advanced Technology Development Project for Predicting and Preventing Chemical Accident" program, funded by Korea Ministry of Climate, Energy, and Environment (MCEE) (RS-2023-00219144).

†Electronic Supplementary Information (ESI) available: [ESI contained A Complete List of the Compounds, further prompts, and details of metrics]. See DOI: 10.5478/MSL.2025.16.4.111

References

1. Jang, I.; Lee, J.-u.; Lee, J.-m.; Kim, B.H.; Moon, B.; Hong, J.; Oh, H.B. *Anal. Chem.* **2019**, 91(14), 9119-9128. https://doi.org/10.1021/acs.analchem.9b01643
2. Lee, S.Y.; Lee, S.T.; Suh, S.; Ko, B.J.; Oh, H.B. *J Anal Toxicol.* **2022,** Aug 13;46(7):732-742. https://doi.org/10.1093/jat/bkab098. PMID: 34498039
3. Choi, E.; Yoo, W.J.; Jang, H.-Y.; Kim, T.-Y.; Lee, S.K.; Oh, H.B. *Journal of Chromatography A*, Volume 1705, **2023**, 464167, ISSN 0021-9673, https://doi.org/10.1016/j.chroma.2023.464167
4. Choi, E.; Choi, Y.; Lee, H.; Kim, J.W.; Oh, H.B. *Bull. Korean Chem. Soc*. **2024**, 45(5), 472. https://doi.org/10.1002/bkcs.12835
5. Lee, S.H.; Choi, E.; Park, J.; Yoon, S.; Song, M.-H.; Lee, J.Y.; Seo, J.; Shin, S.K.; Lee, S.H.; Oh, H.B. *Sci Rep* 15, 18186 (**2025**). https://doi.org/10.1038/s41598-025-02590-y
6. Lowden, C. *Drug Discovery Trends*. **2025**. https://www.drugdiscoverytrends.com/from-data-to-drug-candidates-optimizing-informatics-for-ml-and-genai/
7. Shah, B.; Bleys, J.; Viswa, C, A.; Zurkiya, D.; Leydon, E. *McKinsey & Company*. January 9, **2024**. https://www.mckinsey.com/industries/life-sciences/our-insights/generative-ai-in-the-pharmaceutical-industry-moving-from-hype-to-reality
8. Ayres, L.B.; Gomez, F.J.V.; Linton, J.R.; Silva, M.F.; Garcia, C.D. *Analytica Chimica Acta*, Volume 1161, **2021**, 338403, ISSN 0003-2670, https://doi.org/10.1016/j.aca.2021.338403.
9. Shrivastava, A.D.; Swainston, N.; Samanta, S.; Roberts, I.; Wright Muelas, M.; Kell, D.B. *Biomolecules*. **2021** Nov 30;11(12):1793. https://doi.org/10.3390/biom11121793. PMID: 34944436
10. Hong, Y.; Ye, Y; Tang, H. *Annu. Rev. Anal. Chem*. Volume 18, **2025**. https://doi.org/10.1146/annurev-anchem-071224-082157
11. Whitehead, H.D.; Hayes, K.L.; Swartz, J.A.; Prete, E.; Robison-Taylor, L.; Ellen Mackesy-Amiti, M.; Jimenez, A.D.; Lieberman, M. *Forensic Chem*. **2023** May;33:100475. https://doi.org/10.1016/j.forc.2023.100475.
12. Kalogiouri, N.P.; Aalizadeh, R.; Dasenaki, M.E.; Thomaidis, N.S. *Anal Chim Acta.* **2020** Oct 16;1134:150-173. https://doi.org/10.1016/j.aca.2020.07.029. Epub 2020 Jul 30. PMID: 33059861.
13. Yéléhé-Okouma, M.; Pape, E.; Humbertjean, L.; Evrard, M.; El Osta, R.; Petitpain, N.; Gillet, P.; El Balkhi, S.; Scala-Bertola, J. *Fundam Clin Pharmacol*. **2021** Oct; 35(5):792-807. https://doi.org/10.1111/fcp.12653. Epub 2021 Mar 8. PMID: 33484004.
14. Nielen, M. W. F.; van Engelen, M. C. R.; van de Koppel, S.; Venhuis, B. J.; Zomer, P. Illicit_ED_products_RIVM_report_370030003.pdf. *RIVM Report* 370030003. **2011**. https://www.rivm.nl/bibliotheek/rapporten/370030003.pdf
15. Google Cloud. Supported features for Gemini Code Assist for individuals. **2025**. https://developers.google.com/gemini-code-assist/docs/overview