"Machine learning for functional group identification in vibrational spectroscopy: A pedagogical lab for undergraduate chemistry students"

Elizabeth S. Thrall[1]*, Seung Eun Lee[2], Joshua Schrier[1], Yijun Zhao[2]

[1]Department of Chemistry, Fordham University, The Bronx, NY 10458, United States

[2]Department of Computer and Information Sciences, Fordham University, New York, NY 10023, United States

*To whom correspondence should be addressed

Abstract:

Techniques from the branch of artificial intelligence known as machine learning (ML) have been applied to a wide range of problems in chemistry. Nonetheless, there are very few examples of pedagogical activities to introduce ML to chemistry students in the chemistry education literature. Here we report a computational activity that introduces undergraduate physical chemistry students to ML in the context of vibrational spectroscopy. In the first part of the activity, students use ML binary classification algorithms to distinguish between carbonyl-containing and non-carbonyl-containing molecules on the basis of their infrared absorption spectra. In the second part of the activity, students test modifications to this basic analysis, including different analysis parameters, different ML algorithms, and different test datasets. In a final extension of the activity, students implement a multiclass classification to predict whether carbonyl-containing molecules contain a ketone, a carboxylic acid, or another carbonyl group. This activity is designed to introduce students both to the basic workflow of a ML classification analysis and to some of the ways in which machine learning analyses can fail. We provide a comprehensive handout for the activity, including theoretical background and a detailed protocol, as well as datasets and code to implement the exercise in Python or Mathematica. This activity is designed as a standalone exercise for physical chemistry lab classes but can also be integrated with courses or modules on vibrational spectroscopy and computational chemistry. On the basis of student surveys, we conclude that this activity was successful in introducing students to applications of ML in chemistry.

Main text:

Introduction:

Vibrational spectroscopy is a powerful molecular characterization tool encountered in the organic, analytical, and physical chemistry undergraduate curricula. (Opportunities to incorporate vibrational spectroscopy into the general chemistry curriculum have also been discussed in this Journal.[1–3]) The first introduction to vibrational spectroscopy is often in the context of qualitative analysis, in which students learn to interpret vibrational spectra by looking for the spectral signatures of specific functional groups. A variety of approaches have been discussed for improving student's ability to learn vibrational spectral characteristics, such as inquiry-based card games,[4] physical models,[5] and virtual reality.[6]

The past decade has witnessed a rapid growth of machine learning (ML) for a variety of applications, such as image classification and machine translation. Broadly defined, ML algorithms use example training data to "learn" a model that can be used to make predictions or decisions, without the need for explicit programming of the model. ML has been applied to a wide variety of problems in chemistry[7,8] and efforts have been made to formalize best practices for these studies.[9] More specifically, ML has been applied to both infrared (IR) absorption and Raman vibrational spectroscopy,[10,11] including functional group identification.[12–15]

Although the chemical education community acknowledges the need for student training in computational methods and ML,[16] there are limited pedagogical materials and no standard way of incorporating this into the curriculum. One approach has been the development of dedicated semester-long courses in scientific computing for chemists[17] or cheminformatics[18] that introduce programming in general and include modules on ML methods. There are also dedicated courses on data science for chemistry.[19,20] Another approach is the development of standalone laboratory or classroom experiences that can be incorporated into existing classes, such as prediction of fluid properties in introductory chemical engineering courses[21] or computer vision image analysis to distinguish different types of laboratory glassware.[22] We take the latter approach in this article.

We report a way to introduce ML into the chemistry curriculum by relating it to existing curricular activities pertaining to vibrational spectroscopy. Students immediately grasp the practical value and challenges of deducing molecular structure from spectra. Here we describe a laboratory activity in which students apply ML to functional group identification in IR spectra. Students construct binary classifier models to identify carbonyl-containing compounds from IR spectra (using a database of calculated IR spectra), followed by multiclass classification to distinguish ketones from carboxylic acids or other carbonyls. This activity could be incorporated into many locations in the undergraduate curriculum, but we have designed it for a junior/senior level physical chemistry lecture or lab course, where it can be integrated with existing experimental or computational vibrational spectroscopy lab experiences. Different versions of this experiment can be completed in either one or two 3-hour lab periods, with a number of optional additional components. The computational dataset and the code needed for the experiment are provided as both Python-based Jupyter notebooks and Mathematica notebooks. Interactive notebook-based programming environments have many advantages both for education[17,23–26] and for practicing scientists.[27]

Methods:

Below we provide an overview of the methods used in this study. Additional background is available in the Student Handout and additional technical details are provided in the Supporting Information. The latest versions of the notebooks and student handout are available from GitHub.[28]

Dataset

This experiment uses a dataset from the Alexandria Library containing the vibrational frequencies and intensities for 2,337 molecules calculated using density functional theory (DFT) with the B3LYP hybrid functional and the aug-cc-pVTZ basis set.[29] (Although experimental IR spectra are available from the National Institute of Standards and Technology[30] and other sources, many of these databases are heterogeneous, containing spectra for compounds in different phases with different units, complicating their use.) In the Alexandria Library, each vibrational mode has a calculated vibrational frequency and oscillator strength describing its intensity. To simulate IR absorption spectra, we convolve the vibrational frequencies with a Lorentzian function of 40 cm^{-1} width (Figure 1 and Figure 2A). Of the resulting spectra, 90% (2,104) are used as training data and 10% (233) are used as test data for the binary classification. The total dataset contains spectra of 351 carbonyl-containing molecules, of which 90% (316) are used as training data and 10% (35) are used as test data for the multiclass classification. A Mathematica 12.1 script to perform this data processing is provided in the Supporting Information; in our activity students are provided with the compiled training and test spectra, so they do not need to conduct this step.

Data Preprocessing and Machine Learning

The training and test spectra must be processed further prior to machine learning analysis (Figure 1). First, the spectral intensities for each molecule are normalized from 0 to 1 (Figure 2A and 2B). The intensities in the Alexandria Library represent the oscillator strengths of each vibrational mode, but this step allows analysis of datasets where intensity values may reflect concentration differences or other experimental parameters. Next a thresholding step is performed, in which intensity values below a specified threshold (0.2 by default) are set to 0 (Figure 2C). The data are then split to separate the attributes (*i.e.*, the spectral intensity at each frequency) and the labels (*i.e.*, a 0 or 1 indicating whether a carbonyl is absent or present) as two different variables. Finally, a data balancing step is performed. Carbonyl-containing compounds represent only 15% of the Alexandria Library dataset. Such an imbalance can sometimes cause ML classification models, which often assume uniform distribution of training samples among the classes,[31] to be biased toward the majority class. A data balancing approach called synthetic minority oversampling technique (SMOTE)[32] is used to generate new synthetic instances of the carbonyl class for the training dataset. The resulting training dataset, containing 50% carbonyl-containing and 50% non-carbonyl-containing spectra, is used to train the ML models. Students are guided through each of these data processing steps in the activity.

Four common ML classification algorithms are implemented in this exercise: Decision Tree,[33] Random Forest,[34] *k*-Nearest Neighbors,[35] and Naive Bayes.[36] These algorithms were chosen because they are widely-used, robust, and easy to understand; as described below, they also perform very well for this classification task. (They are not the only choices, and the research literature provides other alternatives, such as support vector machines and neural networks, that may be more appropriate vibrational spectral analysis.[12,13]) The student handout and supporting information provide brief discussions of the assumptions of each of these models. The multiclass classification performed in Part III of this activity is structured as multiple one-vs-all binary

classifications, in which the probability of membership in each class is separately determined for each molecule and the class with the highest probability is then taken as the final predicted label. Standard metrics are used to evaluate model performance, including the accuracy (the proportion of the total number of predictions that were correct), the sensitivity (the proportion of actual positive cases which are correctly identified), and the specificity (proportion of actual negative cases which are correctly identified).

The Python (version 3.7) implementation of this activity uses the scikit-learn library[37] for the ML algorithms, the imbalanced-learn (imblearn) library[38] for SMOTE, and other common libraries for handling datasets (pandas[39]), carrying out mathematical and statistical calculations (NumPy[40] and SciPy[41]), and visualizing data (Matplotlib,[42] Plotly,[43] and Seaborn[44]). Our implementation was performed in the (free-to-use) Google Colaboratory environment,[45] but it could also be run in any standard Jupyter notebook environment. The Mathematica implementation uses built-in functionality available in version 12.1 and above and it includes a custom implementation of the SMOTE algorithm.

Results:

Binary Classification

In Parts I and II of the experiment, students use different binary classification algorithms to predict whether input IR absorption spectra correspond to carbonyl-containing molecules. The IR spectra of carbonyl-containing compounds are characterized by the strong carbonyl stretching mode at 1540 – 1870 cm^{-1} (Figure 2A).[46] The default classification analysis uses a threshold value of 0.2, removing weaker vibrational modes (Figure 2C). Table S1 shows representative performance metrics for the four ML algorithms tested. Although the performance will vary slightly due to randomness in SMOTE balancing and model training, model performance overall was very good, with accuracies, sensitivities, and specificities greater than 95% for the Random Forest and *k*-Nearest Neighbors models and slightly lower for Decision Tree. The Gaussian Naive Bayes model is the clear outlier in performance, with an accuracy of 85%, likely due to its inaccurate assumption of feature independence.

After this basic analysis is implemented, students explore how different analysis parameters affect the binary classification performance. For example, students can assess the effect of changing the threshold setting. There is little change in the performance of the Random Forest model when this setting is decreased to 0.0 from its default value of 0.2 (Table S2). Increasing this setting to a large value, such as 0.5, removes all but the strongest vibrational modes, yet the model performance is still quite good. The sample protocol encourages students to consider the implications regarding what features the model uses to make predictions, and to reflect on similarities and differences in how humans interpret vibrational spectra. Students can also explore the effect of changing the parameters in the different ML algorithms, such as the number of neighbors to use in label prediction for the *k*-Nearest Neighbors algorithm. Moderate increases in this parameter from the default value of five have little effect, but larger increases cause model

performance to deteriorate because the prediction relies on an increasing number of non-similar neighbors (Table S2). Finally, students can run the analysis without SMOTE data balancing; in this case they will observe little change in the performance, indicating that class imbalance is not necessarily a problem if the classes are sufficiently distinct (Table S2). These examples demonstrate that model performance depends on the choice of analysis parameters, and that inappropriate analysis parameters can give poor results.

A goal of this activity is for students to appreciate possible pitfalls and failure modes of ML analyses. Part II prompts students to analyze false positive and false negative error cases for the different ML models. Although there is some variation in these error cases, several common false positives and negatives are shown in Figure 3. Common false positives, such as trans-nitrous acid or chromium dihydride, have a strong IR absorption peak near 1770 – 1780 cm^{-1}, in the same range as the carbonyl stretch. For some of the common false negatives, like *N*,*N*-diethylbutanamide or *o*-tolualdehyde, the carbonyl stretch is shifted to lower frequencies due to electron-donating substituents, which likely explains the failure of the model to classify them as carbonyls. Analysis of these error cases helps students understand that ML models can fail. To illustrate this point further, students are given four spectra of the carbonyl-containing molecule N-methylacetamide, three of which are formatted in various ways inconsistent with the training data; for example, one spectrum uses a smaller spacing between data points so that a smaller frequency range is covered. Students will observe that the models correctly identify the carbonyl group for the correctly formatted spectrum but fail for the incorrectly formatted spectra. This example demonstrates the importance of data preprocessing and the use of consistent data for training and testing ML models.

The sample protocol suggests several other optional extensions of the binary classification task, such as generating a learning curve by repeating the analysis using subsets of the training dataset or using the trained ML models for analysis of other spectra obtained experimentally or computationally. The National Institute of Standards and Technology (NIST) Chemistry WebBook[30] and Computational Chemistry Comparison and Benchmark DataBase (CCCBDB)[47] are good resources for experimental and computational IR absorption spectra, respectively, or students could analyze spectra that they obtained in previous course modules on IR spectroscopy or computational chemistry.

Multiclass Classification

In Part III of the experiment, students implement a multiclass classification model using the Random Forest algorithm to classify carbonyl-containing molecules as ketones, carboxylic acids, or other. Distinguishing between different carbonyl-containing molecules is more challenging than the binary classification analysis, and the model accuracy is correspondingly lower, with an overall accuracy of 80% (Tables S3 and S4). Nevertheless, prediction of carboxylic acids, which have a broad hydroxyl stretching mode in the range of 2500 – 3300 cm^{-1},[46] is fairly accurate, as is prediction of other carbonyl-containing molecules. Ketones, which lack distinctive spectral features other than the carbonyl stretch, have the lowest accuracy (40%). It should be

noted that the training dataset for this part of the analysis is smaller than for the binary classification, which likely decreases the overall model performance.

Although the sample protocol does not provide possible extensions, Part III could be expanded by using different ML algorithms, analysis parameters, or datasets. Students could also analyze other carbonyl-containing functional groups, such as amides or aldehydes, although there are fewer instances of these groups in the Alexandria Library dataset.

Implementation:

This activity was implemented in the Spring 2021 semester in physical chemistry lab courses at Fordham University and Whitman College, with a total of 22 junior and senior chemistry majors. The core exercise is designed so that Parts I and II can be carried out in a single 3 – 4 hour laboratory period and Part III can be carried out in a second period of the same duration. Student estimates of time required to complete the activity were consistent with this timeline (Figure 4), with most students requiring one hour or less to complete Part I and approximately two hours each to complete Parts II and III. If two laboratory periods are not available, Parts I and II can be implemented as a one-day activity that still gives students an overview of ML classification tasks and possible pitfalls. The exercise was carried out asynchronously at Fordham University and synchronously but remotely at Whitman College; completion times may be shorter in an in-person setting where the instructor would be able to assist students more easily.

This exercise is implemented in two notebooks, one for the binary classification in Parts I and II and a second for the multiclass classification in Part III. These notebooks provide explanatory information and computer code needed to carry out the analysis. The first notebook contains the complete code needed to carry out binary classification in Part I of the activity. It can then be modified by students to carry out the additional analyses in Part II. The second notebook, used for the multiclass classification task in Part III, only contains a framework for the analysis. Students must write larger chunks of code on their own, all of which can be adapted from the binary classification notebook, to complete the analysis. In this way the three parts of the exercise are designed so that students move from simply executing code and observing the output, to making small changes to code, and finally to writing their own larger blocks of code. These notebooks are available as both Python and Mathematica notebooks. Python notebooks can be executed with the free web-based Google Colaboratory platform,[45] or using any available Jupyter Notebook environment. Mathematica notebooks require a license for either the desktop or online version of Mathematica.

Although our trials at Fordham University and Whitman College used this activity as a standalone exercise, instructors could integrate this activity with other course modules and experiments. For example, this exercise could be performed after a course module on experimental IR absorption spectroscopy, allowing students to use their trained ML models to classify their experimentally-obtained spectra. Alternatively, this activity could form the basis of a larger exercise or independent project in a computational chemistry or machine learning course. Students could extend the multiclass classification model in Part III to include other carbonyl-containing

functional groups, or they could use the same approach to train a multiclass classifier to distinguish between carbonyls and other functional groups (such as alkenes, alcohols, or amines).

Results of Student Surveys:

Anonymous surveys were conducted before and after the experiment to assess students' previous experience with ML and Python programming and to determine students' assessment of the effectiveness of the exercise as an introduction to these topics. The pre-lab survey (Figure 5) revealed that although most students had heard the term "machine learning" (71%) and were familiar with its non-chemistry applications (57%), only a small fraction (23%) knew of its chemistry applications. About half of students (48%) had taken at least one computer science course previously, either in high school or in college. Almost all students (95%) had heard of Python, with smaller fractions having read (57%) or written (43%) Python code, but almost no one had previous experiments working in Google Colaboratory or another Python notebook environment (5%).

The post-lab survey (Figure 6) demonstrated that students found the experiment to be an effective introduction to ML, with large majorities agreeing or strongly agreeing that after the exercise they understood the basic steps in a ML classification task (95%), some of the factor that affect ML analyses and possible pitfalls (94%), and some of the possible applications of ML to chemistry (84%). Most students felt more able to work in a Google Colaboratory environment (84%) and to read Python code (68%), although not to write Python code (26%). Finally, most students enjoyed the exercise (63%) and reported increased interest in learning more about ML (74%). From these results, we conclude that the activity functions as an effective introduction to ML and to working with Python code in a Google Colaboratory environment, even if it is not a comprehensive introduction to the Python programming language.

Conclusion:

We have developed an activity in which students train ML algorithms to distinguish carbonyl-containing compounds from IR absorption spectra. This activity, although designed for a physical chemistry laboratory class, can be incorporated into other chemistry or programming courses. It can be customized to align with other course activities and can provide a foundation for more advanced independent projects. Surveys reveal that this activity was effective both in introducing students to applications of ML in chemistry and in stimulating student interest in the topic. In light of the growing importance of computational methods and artificial intelligence across the chemical sciences, activities such as these are an important component of a modern education in chemistry.

Acknowledgments:

We thank Mark Hendricks for testing this experiment in his physical chemistry lab course, Kedan He for helpful comments on the materials, and the physical chemistry lab students at Fordham University and Whitman College for their participation in and feedback on the

experiment. Funding for this project was provided by the Fordham University Graduate School of Arts and Sciences through the Professional Development Grant program, and the National Science Foundation (DMR-1928882) and the Henry Dreyfus Teacher-Scholar Award (TH-14-010).

References:

(1) Hill, M. A. Infrared Spectroscopy in the General Chemistry Lab. *J. Chem. Educ.* **2001**, *78* (1), 26. https://doi.org/10.1021/ed078p26.

(2) Heuer, W. B.; Koubek, E. An Investigation into the Absorption of Infrared Light by Small Molecules: A General Chemistry Experiment. *J. Chem. Educ.* **1997**, *74* (3), 313. https://doi.org/10.1021/ed074p313.

(3) Csizmar, C. M.; Force, D. A.; Warner, D. L. Examination of Bond Properties through Infrared Spectroscopy and Molecular Modeling in the General Chemistry Laboratory. *J. Chem. Educ.* **2012**, *89* (3), 379–382. https://doi.org/10.1021/ed200100n.

(4) Bennett, J.; Forster, T. IR Cards: Inquiry-Based Introduction to Infrared Spectroscopy. *J. Chem. Educ.* **2010**, *87* (1), 73–77. https://doi.org/10.1021/ed800009k.

(5) Wright, L. C.; Oliver-Hoyo, M. T. Supporting the Teaching of Infrared Spectroscopy Concepts Using a Physical Model. *J. Chem. Educ.* **2019**, *96* (5), 1015–1021. https://doi.org/10.1021/acs.jchemed.8b00805.

(6) Dunnagan, C. L.; Dannenberg, D. A.; Cuales, M. P.; Earnest, A. D.; Gurnsey, R. M.; Gallardo-Williams, M. T. Production and Evaluation of a Realistic Immersive Virtual Reality Organic Chemistry Laboratory Experience: Infrared Spectroscopy. *J. Chem. Educ.* **2020**, *97* (1), 258–262. https://doi.org/10.1021/acs.jchemed.9b00705.

(7) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559* (7715), 547–555. https://doi.org/10.1038/s41586-018-0337-2.

(8) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; American Chemical Society: Washington, DC, USA, 2020. https://doi.org/10.1021/acs.infocus.7e4001.

(9) Artrith, N.; Butler, K. T.; Coudert, F.-X.; Han, S.; Isayev, O.; Jain, A.; Walsh, A. Best Practices in Machine Learning for Chemistry. *Nat. Chem.* **2021**, *13* (6), 505–508. https://doi.org/10.1038/s41557-021-00716-z.

(10) Lussier, F.; Thibault, V.; Charron, B.; Wallace, G. Q.; Masson, J.-F. Deep Learning and Artificial Intelligence Methods for Raman and Surface-Enhanced Raman Scattering. *TrAC Trends in Analytical Chemistry* **2020**, *124*, 115796. https://doi.org/10.1016/j.trac.2019.115796.

(11) Yang, J.; Xu, J.; Zhang, X.; Wu, C.; Lin, T.; Ying, Y. Deep Learning for Vibrational Spectral Analysis: Recent Progress and a Practical Guide. *Analytica Chimica Acta* **2019**, *1081*, 6–17. https://doi.org/10.1016/j.aca.2019.06.012.

(12) Fine, J. A.; Rajasekar, A. A.; Jethava, K. P.; Chopra, G. Spectral Deep Learning for Prediction and Prospective Validation of Functional Groups. *Chem. Sci.* **2020**, *11* (18), 4618–4630. https://doi.org/10.1039/C9SC06240H.

(13) Wang, Z.; Feng, X.; Liu, J.; Lu, M.; Li, M. Functional Groups Prediction from Infrared Spectra Based on Computer-Assist Approaches. *Microchemical Journal* **2020**, *159*, 105395. https://doi.org/10.1016/j.microc.2020.105395.

(14) Nalla, R.; Pinge, R.; Narwaria, M.; Chaudhury, B. Priority Based Functional Group Identification of Organic Molecules Using Machine Learning. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*; ACM: Goa India, 2018; pp 201–209. https://doi.org/10.1145/3152494.3152522.
(15) Enders, A.; North, N.; Fensore, C.; Velez-Alvarez, J.; Allen, H. *Functional Group Identification for FTIR Spectra Using Image-Based Machine Learning Models*; preprint; 2021. https://doi.org/10.26434/chemrxiv.14188679.v1.
(16) Holme, T. A. Can Today's Chemistry Curriculum Actually Produce Tomorrow's Adaptable Chemist? *J. Chem. Educ.* **2019**, *96* (4), 611–612. https://doi.org/10.1021/acs.jchemed.9b00112.
(17) Weiss, C. J. A Creative Commons Textbook for Teaching Scientific Computing to Chemistry Students with Python and Jupyter Notebooks. *J. Chem. Educ.* **2021**, *98* (2), 489–494. https://doi.org/10.1021/acs.jchemed.0c01071.
(18) Kim, S.; Bucholtz, E. C.; Briney, K.; Cornell, A. P.; Cuadros, J.; Fulfer, K. D.; Gupta, T.; Hepler-Smith, E.; Johnston, D. H.; Lang, A. S. I. D.; Larsen, D.; Li, Y.; McEwen, L. R.; Morsch, L. A.; Muzyka, J. L.; Belford, R. E. Teaching Cheminformatics through a Collaborative Intercollegiate Online Chemistry Course (OLCC). *J. Chem. Educ.* **2021**, *98* (2), 416–425. https://doi.org/10.1021/acs.jchemed.0c01035.
(19) Gressling, T. *Data Science in Chemistry: Artificial Intelligence, Big Data, Chemometrics and Quantum Computing with Jupyter*; De Gruyter, 2020. https://doi.org/10.1515/9783110629453.
(20) Lafuente, D.; Cohen, B.; Fiorini, G.; García, A.; Bringas, M.; Morzan, E.; Onna, D. *Introduction to Machine Learning for Chemists: An Undergraduate Course Using Python Notebooks for Visualization, Data Processing, Data Analysis, and Data Modeling*; preprint; 2021. https://doi.org/10.26434/chemrxiv.13749199.v1.
(21) Joss, L.; Müller, E. A. Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB. *J. Chem. Educ.* **2019**, *96* (4), 697–703. https://doi.org/10.1021/acs.jchemed.8b00692.
(22) Sharma, A. K. Laboratory Glassware Identification: Supervised Machine Learning Example for Science Students. *JOCSE* **2021**, *12* (1), 8–15. https://doi.org/10.22369/issn.2153-4136/12/1/2.
(23) Engelberger, F.; Galaz-Davison, P.; Bravo, G.; Rivera, M.; Ramírez-Sarmiento, C. A. Developing and Implementing Cloud-Based Tutorials That Combine Bioinformatics Software, Interactive Coding, and Visualization Exercises for Distance Learning on Structural Bioinformatics. *J. Chem. Educ.* **2021**, *98* (5), 1801–1807. https://doi.org/10.1021/acs.jchemed.1c00022.
(24) Barba, L. A. Engineers Code: Reusable Open Learning Modules for Engineering Computations. *Comput. Sci. Eng.* **2020**, *22* (4), 26–35. https://doi.org/10.1109/MCSE.2020.2976002.
(25) Menke, E. J. Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course. *J. Chem. Educ.* **2020**, *97* (10), 3899–3903. https://doi.org/10.1021/acs.jchemed.9b01131.
(26) Verrett, J.; Boukouvala, F.; Dowling, A.; Ulissi, Z.; Zavala, V. Computational Notebooks in Chemical Engineering Curricula. *Chem Eng Educ* **2020**, *54* (3), 143–150.
(27) Perkel, J. M. Reactive, Reproducible, Collaborative: Computational Notebooks Evolve. *Nature* **2021**, *593* (7857), 156–157. https://doi.org/10.1038/d41586-021-01174-w.

(28) https://github.com/elizabeththrall/MLforPChem/tree/main/MLforvibspectroscopy (accessed 2021-06-18).
(29) Ghahremanpour, M. M.; van Maaren, P. J.; van der Spoel, D. The Alexandria Library, a Quantum-Chemical Database of Molecular Properties for Force Field Development. *Sci Data* **2018**, *5* (1), 180062. https://doi.org/10.1038/sdata.2018.62.
(30) NIST Chemistry WebBook. https://webbook.nist.gov (accessed 2021-06-03).
(31) Chawla, N. V.; Japkowicz, N.; Kotcz, A. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explor. Newsl.* **2004**, *6* (1), 1–6. https://doi.org/10.1145/1007730.1007733.
(32) Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. https://doi.org/10.1613/jair.953.
(33) Quinlan, J. R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1* (1), 81–106. https://doi.org/10.1007/BF00116251.
(34) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5–32. https://doi.org/10.1023/A:1010933404324.
(35) Altman, N. S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *null* **1992**, *46* (3), 175–185. https://doi.org/10.1080/00031305.1992.10475879.
(36) Al-Aidaroos, K. M.; Bakar, A. A.; Othman, Z. Naive Bayes Variants in Classification Learning. In *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*; IEEE: Shah Alam, Selangor, 2010; pp 276–281. https://doi.org/10.1109/INFRKM.2010.5466902.
(37) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12* (85), 2825–2830.
(38) Lemaître, G.; Nogueira, F.; Aridas, C. K. Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18* (1), 559–563.
(39) McKinney, W. Pandas: A Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing* **2011**, *14* (9), 1–9.
(40) Oliphant, T. E. *A Guide to NumPy*; Trelgol Publishing: USA, 2006; Vol. 1.
(41) SciPy 1.0 Contributors; Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat Methods* **2020**, *17* (3), 261–272. https://doi.org/10.1038/s41592-019-0686-2.
(42) Barrett, P.; Hunter, J. D.; Miller, T.; Hsu, J. Matplotlib -- A Portable Python Plotting Package. In *Astronomical data analysis software and systems XIV*; 2005; Vol. 347, p 91.
(43) Plotly Python Graphing Library. https://plotly.com/python (accessed 2021-06-03).
(44) Waskom, M. Seaborn: Statistical Data Visualization. *JOSS* **2021**, *6* (60), 3021. https://doi.org/10.21105/joss.03021.
(45) Google Colaboratory. https://colab.research.google.com (accessed 2021-06-03).

(46) Silverstein, R. M.; Webster, F. X.; Kiemle, D. J. *Spectrometric Identification of Organic Compounds*, 7th ed.; John Wiley & Sons: Hoboken, NJ, 2005; Chapter 2.6, pp. 92–95.
(47) Computational Chemistry Comparison and Benchmark DataBase. https://cccbdb.nist.gov (accessed 2021-06-03).

Figures:

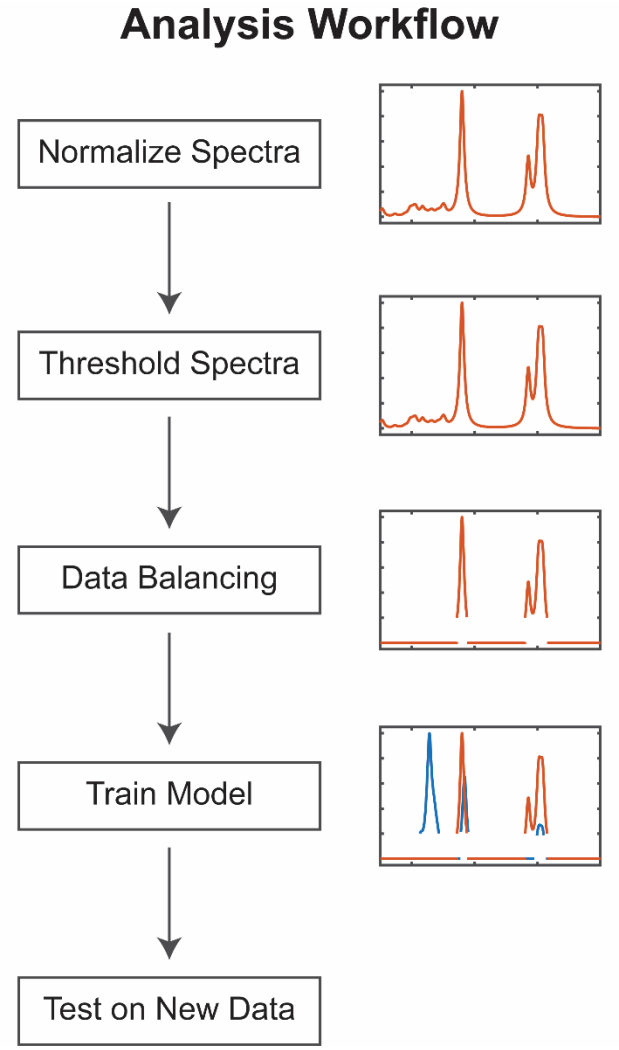

Figure 1. Overview of data preprocessing and machine learning workflow.

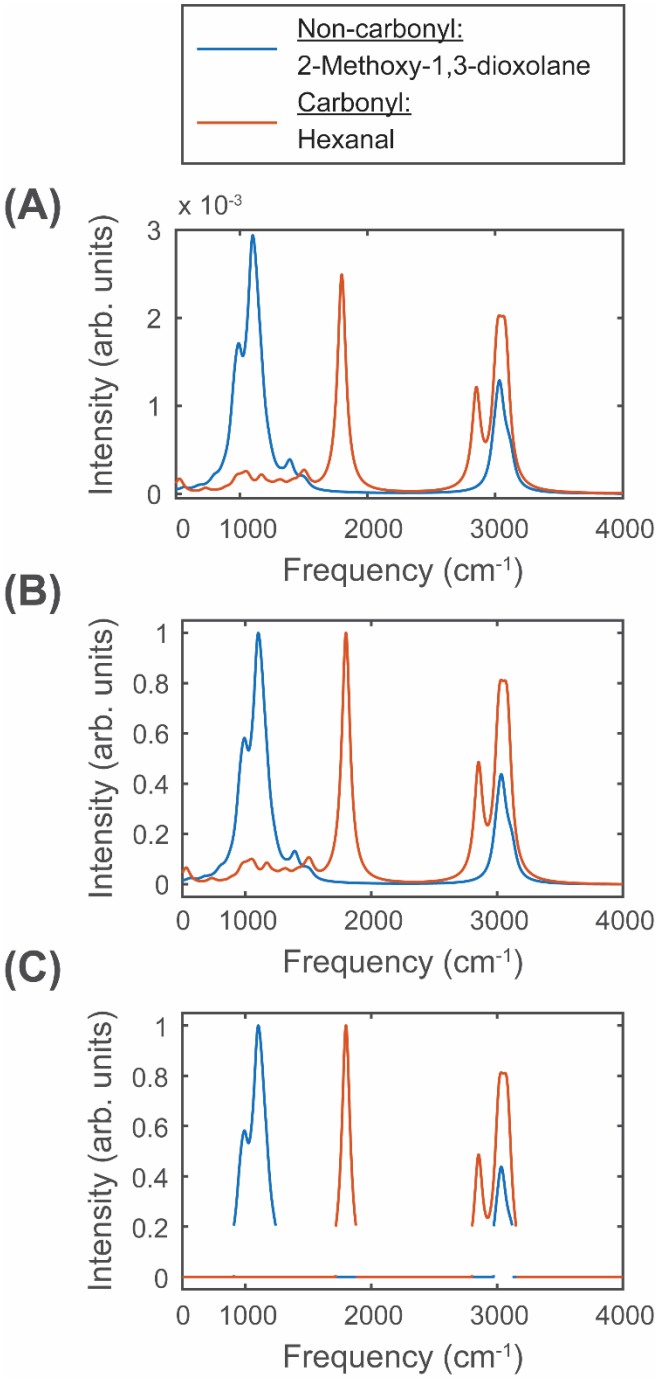

Figure 2. Representative spectra of carbonyl-containing and non-carbonyl-containing molecules. (A) Original. (B) After normalization. (C) After thresholding.

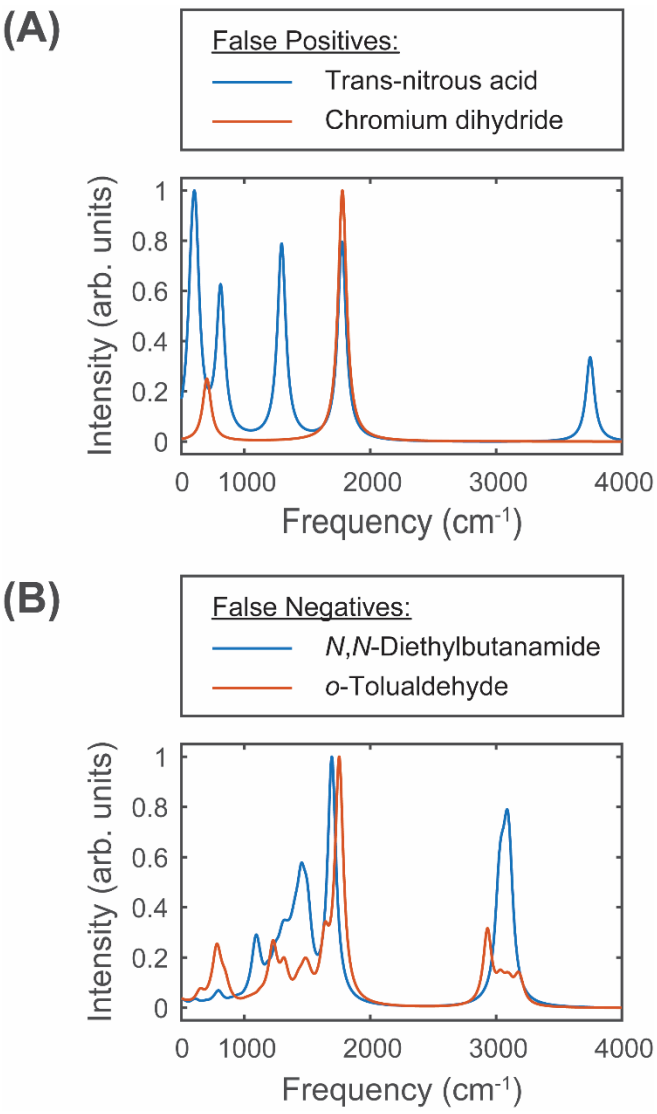

Figure 3. Spectra of (a) common false positives and (b) common false negatives in binary classification analysis.

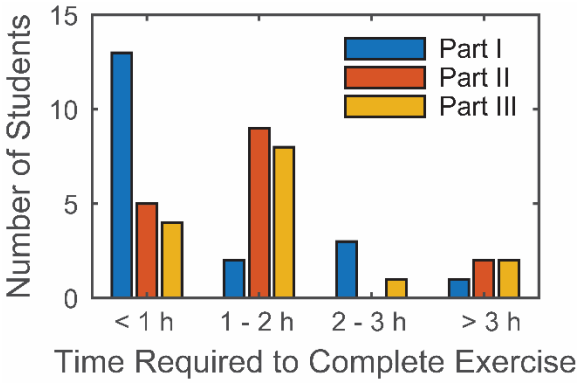

Figure 4. Student estimates of time required to complete Parts I, II, and III of the exercise.

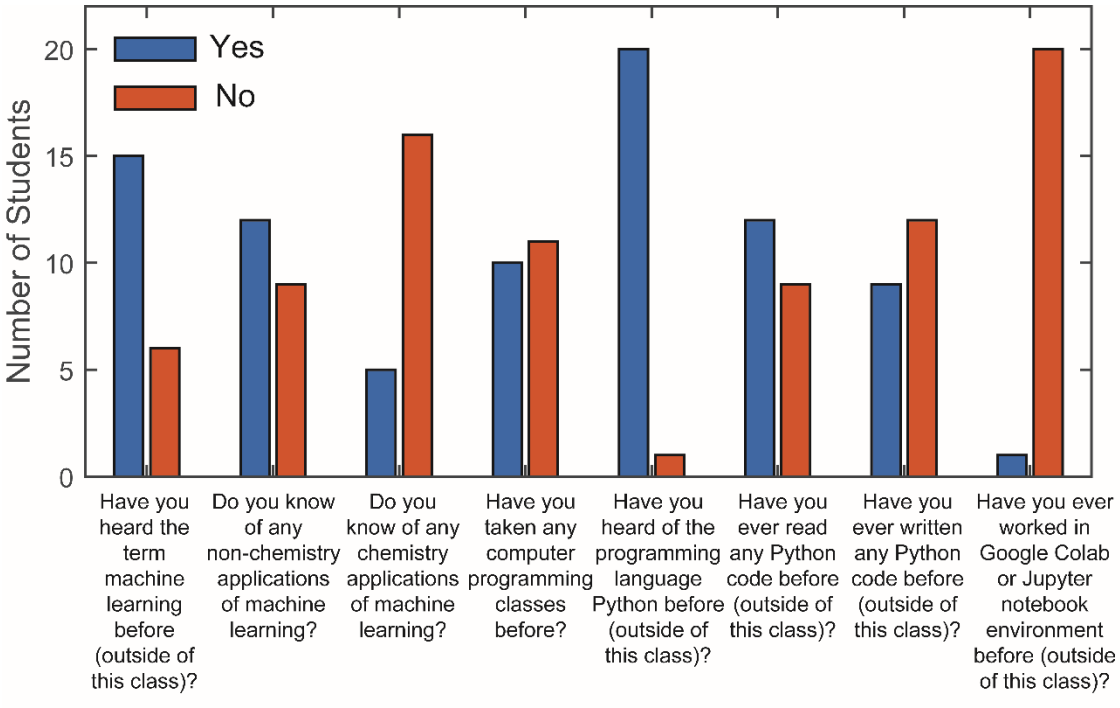

Figure 5. Results of the pre-lab survey assessing students' familiarity with machine learning and Python programming.

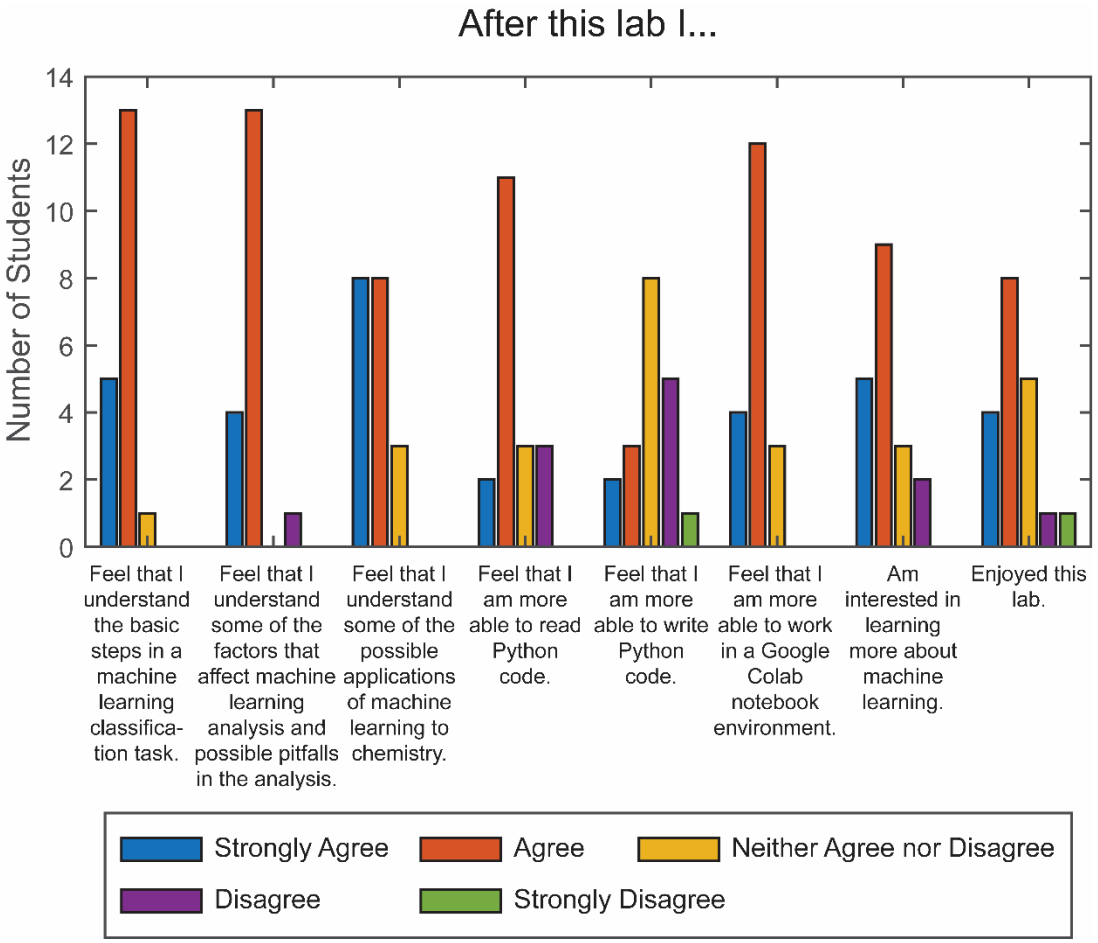

Figure 6. Results of the post-lab survey assessing the effectiveness of the activity in introducing students to machine learning and Python programming and in stimulating student interest in machine learning.