The Journal of Open Source Education

Machine Learning in Geosciences

Marine Denolle [1¶], **Nicoleta Cristea** [1], **Akshay Mehra** [1], **Arianne Ducellier** [1], **Ziheng Sun** [2], **Stefan Todoran** [1], **Scott Henderson** [1], and **Claire Jensen** [1]

1 University of Washington, Seattle, USA **2** George Mason University, George Mason, USA **¶** Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 29 January 2025
Published: unpublished

Summary

The “Machine Learning in the Geosciences” course—which has been offered as ESS 469/569 at the University of Washington since 2023— introduces undergraduate and graduate students to the use of machine learning (ML) techniques within a geoscientific context.

Statement of need

Machine learning (ML) has rapidly emerged as a transformative tool in the analysis of big data and scientific discovery across disciplines, especially since 2010. Geosciences, with its inherently large, complex, and multidimensional datasets, is particularly poised to benefit from ML’s capabilities Mousavi & Beroza (2022). Yet, despite the explosion of ML applications in geoscientific research, there is no established curriculum in higher education that focuses on equipping students with practical ML skills tailored to the unique needs of geosciences. Many textbooks are dedicated to statistical learning without geoscience applications(Petrelli, 2021, p. wang2023data).

Generalized data science courses lack the domain-specific emphasis critical for addressing the challenges of geoscientific datasets, such as handling spatiotemporal structures, working with geospatial data formats optimized for cloud systems, addressing variable data quality, and integrating physical constraints into ML models. A course explicitly dedicated to ML in geosciences can bridge this gap, ensuring students and researchers gain the expertise required to tackle pressing environmental and Earth system challenges through ML-driven approaches. ESS 469/569 (Machine Learning in the Geosciences) is such a course.

The **Jupyter Book** created for ESS 469/569 is particularly timely. Geoscience programs across institutions increasingly recognize the critical importance of Artificial Intelligence (AI) and ML research. However, these programs often lack the resources or infrastructure to develop practical, cutting-edge ML curricula independently. Our Jupyter Book provides an accessible, open-source, and modular framework that can easily be integrated into academic programs, accelerating the adoption of AI technologies within geoscientific education and research.

How this course was developed

The course arose from merging an in-development course, “Data Sciences in the Earth and Planetary Sciences” (2021), with an NSF-funded project, Geosmart (Cristea et al., 2024). The result was a senior undergraduate and graduate level course designed for students at the University of Washington who are primarily enrolled in the departments of Earth and Space Sciences, Atmospheric and Climate Sciences, Oceanography, Forestry, Fisheries, and Civil Environmental Engineering. Students in these departments are increasingly interested in applying ML methods to large, complex datasets (for example, climate model outputs that do not fit within the available RAM or hard drive space of personal computers). In 2023, the course was reviewed by colleagues in the Departments of Applied Mathematics and Computer Sciences at the University of Washington to differentiate between “applied machine learning” and “fundamental machine learning.” The course is now offered yearly and enrolls 35-40 students.

Course Structure: ESS 469/569 has three pillars:

1. **AI-ready GeoData:** Focuses on geoscientific data modalities, characteristics, feature extraction, dimensionality reduction, and preparing datasets for AI applications.
2. **Classic Machine Learning:** Covers model training, evaluation, and robust training practices for algorithms such as K-means, random forests, and k-nearest neighbors.
3. **Deep Learning:** Explores foundational deep learning concepts including, but not limited to convolutional neural networks, fully connected layers, sequence-to-sequence learning with recurrent neural networks, and modern topics like physics-informed neural networks and network architecture search.

Technical Skills Development:

The course emphasizes building competencies in:

- **Shell scripting**
- **Version control with Git and GitHub**
- **Generative AI (GenAI)**, integrating GenAI for software development and literature synthesis.
- **Python programming**, utilizing packages such as NumPy, Pandas, scikit-learn, PyTorch
- **Data visualization** using Matplotlib, seaborn, Plotly
- **High-performance computing** strategies for cloud and HPC

Prerequisites:

Students are expected to have completed courses in mathematics, applied mathematics, and statistics. Additionally, students should have completed an intermediate-level programming coursework. While prior knowledge of Python is recommended, the course provides refreshers on computing as needed.

Learning objectives

By the end of the course, students can:

- Demonstrate proficiency in Python programming, Jupyter notebooks, Git version control, integration of GenAI in coding practices (e.g., GitHub Copilot), Conda environments, containerization, and deploying software on new platforms.
- Construct a standard ML workflow that follows community best practices for data preparation, model design, training, validation, and evaluation.
- Implement data manipulation strategies pertinent to geosciences, such as handling time series and spatial information, visualization, dimensionality reduction, and feature engineering.
- Understand and apply open science principles, ensuring reproducibility and adherence to digital scholarship standards.

- Gain familiarity with canonical examples of ML across various geoscience disciplines (e.g., automating data analysis pipelines in seismology to detect earthquakes, multivariate regressions to predict climate and oceanographic variables) and identify strategies for using ML in geoscience in the context of data richness, physical models, and problem setup.
- Evaluate the robustness of the ML pipelines utilized in the scientific literature

An instructor can cover the material in the course book (see below) over approximately 50 hours of instructional hours.

Teaching materials

The class alternates between Jupyter notebooks, slides, and student-led presentations.

Slides

The majority of the class can be taught by going through notebooks in the book. Additionally, we have built several slide decks for the convenience of the instructor. The course GitHub contains raw materials for future instructors to adapt.

- Introduction class: overview of ML in the geosciences, scientific concepts, course logistics. Slides are provided in PPTX format given the dynamic content of introductions for ML in the field.
- Computing Platform a slide deck to support an introduction to the course with resources for literature review, cyberinfrastructure including cloud computing, and a brief motivation to introduce version control.
- Data Definition: an overview of data definition and formats for geosciences to support Chapter 2.1 and 2.2.
- Visualization: an overview of best practices for data and model visualization to supplement the early lectures of Chapter 2.
- AI-ready Dataset are review of what constitutes an AI-ready data set to give at the end of Chapter 2.
- Classification and Regression a slide deck to introduce classification and regression to give at the beginning of Chapter 3.

Small Geoscientific Datasets

We have assembled a small collection of geosciences datasets (total size of approximately 300 MB) for use in both the book and in instruction. These datasets can be found in the GitHub repository MLGEO-data (https://github.com/UW-MLGEO/MLGeo-dataset), which contains notebooks (`./scripts/`) that demonstrate how to source and/or manipulate data.

```
git clone https://github.com/UW-MLGEO/MLGeo-dataset
```

Docker Base Container

We have created a minimal Docker image to run the notebooks in class. This image is automatically built using a GitHub action from this repository (https://github.com/UW-MLGEO/MLGeo-image).

The image can be pulled with Docker:

```
docker pull uwessds/mlgeo-image:latest
```

Technology Integration

Our course emphasizes building a robust technological foundation for students to succeed in applying machine learning in geosciences. In the first week, students are introduced to generative AI (genAI) tools for coding, such as GitHub Copilot, to accelerate their ability to draft and refine code efficiently. A significant focus is placed on ensuring students have access to appropriate software platforms, including setting up VSCode, creating GitHub accounts, and installing either a pre-configured Docker image or a Conda environment tailored for the course. We guide students to help them establish a well-organized workspace, integrating VSCode with Copilot for seamless AI-assisted coding. These “setup” sessions also cover best practices for managing environments, troubleshooting installations, and maintaining reproducibility in their workflows.

Students were allowed and encouraged to use CoPilot for their own homework and projects, and asked to use ChatGPT for self-evaluation and improvements, and demonstrate the outcome of interacting with genAI for evaluation (which highlighted the benefits and flaws of the systems). The integration of genAI provides students with literacy and awareness of both the benefits and potential pitfalls of genAI.

We have also started to use genAI to craft novel geosciences-inspired synthetic data sets for in-class exercises.

Jupyter Book

The MLGEO book is presented as a collection of Jupyter notebooks organized into a Jupyter Book. This format allows for an interactive learning experience, where students can run code cells, visualize data, and experiment with different machine learning models directly within the notebooks.

The Jupyter Book is hosted online and can be accessed through the following link: MLGEO Jupyter Book. Each chapter is divided into multiple sections, with detailed explanations, code examples, and exercises to reinforce the concepts covered.

Content Delivery

The course is structured to provide a balanced and engaging learning experience, with each week designed to focus on three key components: 1/3 conceptual understanding, 1/3 application through toy problems, and 1/3 hands-on student-led exercises.

Weekly student participation includes presenting summaries of scientific papers or webinars to encourage peer learning and collaborative discussions. We have built assignments that can be tackled in groups to align with an equal split between data curation, CML, and deep learning techniques.

Students are provided at least 20 minutes to practice during class, fostering collaborative problem-solving skills through real-time feedback between students. With its reliance on digital tools like Jupyter notebooks, GitHub, and cloud computing platforms, the course is well-suited for remote delivery.

Homework

To reinforce the concepts that we discuss in class, we have designed several assignments for students.

Final Project

Details about the final project, which is group-based (2-4 students), can be found in the course book. An example of such a project is shown in Chapter 7.

Teaching experience

Instructors and students have access to a Jupyter Hub provisioned by the University of Washington for the class, which uses the `uwessds/mlgeo-image` Docker Image for a common computing environment. In the 2024 course offering, we made the students install their environment locally with Visual Studio Code, a student license for GitHub education that included a free license to GitHub CoPilot, and integrated this to the instructional time. Students cloned the Jupyter Book repository on their local Mac, Linux, and PC laptops, and ran the notebooks locally. It took a full week to have all 35 students fully ready to run the notebooks.

The integration of genAI in the 2024 course offering was transformative: the instructor spent less time debugging in class and more time discussing ML concepts, while the students spent less time stuck on software engineering and formatting and more time discussing their data. This acceleration enabled students to complete all four pillars of the final project.

Examples of final projects are shown in Chapter 7

Conclusion and Outlook

Overall, the enhanced teaching experience fostered a more interactive and productive classroom environment, ultimately leading to a more comprehensive understanding of machine learning principles and their practical applications.

The Jupyter book is designed to be a dynamic document to which the community is invited to contribute.

Future improvements should include more geoscientific toy data sets, refinements of statistical learning between univariate and multivariate data, development of student-led exercises, and additional homework.

Acknowledgments

We acknowledge the UW eScience Institute's support provided through office hours and support for the GeoSMART project. Part of this project was supported by the College of the Environment and NSF GeoSMART (GeoScience Machine Learning Resources and Training), award number OAC-2117834. Additional use cases and training resources are available on the GeoSMART website

References

Cristea, N., Sun, Z., Arendt, A., Henderson, S., Denolle, M., & Burgess, A. (2024). *GeoSMART: Machine Learning Training and Curriculum Development for Earth Science Studies*. https://doi.org/10.6084/m9.figshare.26800498.v1

Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in Geophysics*, *61*, 1–55. https://doi.org/10.1016/bs.agph.2020.06.001

Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H. A., & Kumar, V. (2018). Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, *31*(8), 1544–1554. https://doi.org/10.1109/TKDE.2018.2861006

Mousavi, S. M., & Beroza, G. C. (2022). Deep-learning seismology. *Science*, *377*(6607), eabm4470. https://doi.org/10.1126/science.abm4470

Petrelli, M. (2021). *Introduction to python in earth science data analysis: From descriptive statistics to machine learning*. Springer Nature. https://doi.org/10.1007/978-3-030-74859-3

Sun, Z., Sandoval, L., Crystal-Ornelas, R., Mousavi, S. M., Wang, J., Lin, C., Cristea, N., Tong, D., Carande, W. H., Ma, X., & others. (2022). A review of earth artificial intelligence. *Computers & Geosciences*, *159*, 105034. https://doi.org/10.1016/j.cageo.2022.105034