JOSS
The Journal of Open Source Software

Machine Learning in Geosciences

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

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

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished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).

New programs are dedicated to data sciences. The Colorado University- Boulder Earth Data Science Program provides a suite of free online tutorials for data science, especially targetting Python programming, time series data with low sampling rates typically stored in `datetime` objects, geospatial data typical to remote sensing.the University of California - Santa Barbara Master's in Environmental Data Science focuses on data science skills, python skills, and geospatial statistical methods.

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 **JupyterBook** 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 independently develop practical, cutting-edge ML curricula. Our JupyterBook 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.

By offering hands-on, practical experience with ML techniques using geoscientific examples, our course ensures that students not only understand ML concepts but can also directly apply them to real-world problems. This foundational training is vital for preparing the next generation of geoscientists to leverage AI for critical discoveries, from climate change mitigation to natural hazard forecasting to resource exploration.

In summary, ESS 469/569 addresses a growing need in higher education by filling a critical gap in geoscientific training. It equips students with ML expertise, fosters interdisciplinary innovation, and ensures geoscientific programs remain at the forefront of scientific discovery in the era of AI.

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, 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 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, multi-variate 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.

Detailed syllabus

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. Like all public repositories, the course GitHub contains raw materials for future instructors to adapt.

- Introduction class: overview of ML in the geosciences, scientific concepts, course logistics.
-

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 to 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. By mastering such tools early in the course, students are empowered to tackle coding challenges with confidence and efficiency, leveraging cutting-edge AI technologies to enhance their productivity and technical skills.

Students were allowed and encourage to use CoPilot for their own homework and projects, and asked to use chatGPT for self-evaluation and improvements, and demonstrates the outcome of interacting with genAI for evaluation (which highlighted the benefits and flaws of the systems). The integration of genAI overall gives students literacy and awareness of positive and pitfalls of genAI.

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

JupyterBook

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.

The outline of the book is * Chapter 1: Getting Started * Chapter 2: Data Manipulation * Data definition, modalidies, data structure (data frames, arrays) * Statistical analysis for univariate or multivariate data * Data transforms and filtering * Feature engineering * Synthetic Noise * AI/ML-ready data sets * Chapter 3: Machine Learning * Foundamentals of ML: modes of supervisions, classification vs regression, data prep (train, val, test), robustness and generalization * Clustering (unsupervised and supervised) * Classifications * Regression * AutoML * Chapter 4: Deep Learning * Introduction to DL * Training Neural Networks * Classification, Regression, Time series forecast * Popular Model architectures (NN, MLP, CNN, RNN, auto-encoder) * Frontier topics: Neural Architecture Search, PINNS, Large Language Models * Chapter 5: Model Workflows * Discussion about ML full stack reproducibility and Geoweaver * Chapter 6: Cloud Computing * pointers to cloud computing tutorials, with a terraform example and a AWS example * Chapter 7: Use Cases * Collection of projects from the previous course offerings

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. This structure ensures that students not only grasp the theoretical aspects of machine learning but also apply them in practical scenarios and take an active role in the learning process.

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. Homework assignments help instructors assess individual learning outcomes, ensuring students comprehensively understand the materials.

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. However, successful remote implementation requires additional teaching assistants (TAs) and breakout room support to address diverse student needs effectively.

Homework

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

The "Classic Machine Learning" homework (CML) assignment, for example, is designed to reinforce students' understanding of key machine learning concepts introduced in Chapter 3 of the course. The primary objective of the homework is to provide hands-on experience in data preparation, unsupervised clustering, and the application of various supervised learning algorithms.

In the initial phase of the assignment, students engage in data preparation, which includes reading, cleaning, exploring, and reducing the dimensionality of a dataset. This process ensures that students can effectively handle real-world geoscientific data, making it suitable for machine learning applications. Subsequently, students apply unsupervised clustering techniques, (specifically K-means), to identify patterns within the data. This step emphasizes the importance of selecting optimal cluster numbers and evaluating clustering performance.

The assignment culminates with the implementation of various supervised learning models, such as K-Nearest Neighbors, Naive Bayes, Random Forest, Support Vector Machine, and Multi-Layer Perceptron. Students are tasked with feature scaling, splitting data into training and testing sets, designing models, and evaluating their performance using metrics like confusion matrices and cross-validation. This comprehensive approach ensures that students gain practical skills in model selection, training, and evaluation, directly applying the theoretical concepts covered in Chapter 3.

Final Project

The final project, which is group-based (2-4 students), has 4 pillars:

1. Students should **design** a scientifically-sound ML approach =, which includes a justification for the use of ML. Students should also determine the best non-ML approach to solving the problem and use that as a baseline for evaluation.
2. Students should **develop an AI/ML-ready dataset**. To do so, students must:
   - Explore the data (e.g., its dimensionality).
   - Establish a data pipeline.
   - Curate a dataset for ML ingestion.
3. Students begin by creating a **baseline ML using CML** techniques. Students are encouraged to leverage auto-ML to find an optimal model solution.
4. Students should then explore **DL models** and their architectures. If a DL approach improves upon the CML outcomes, then students should set up a comprehensive comparison and argue for the adoption of one approach over the other.

Details about the final project can be found in the course book. Example of such a project is shown in Ch

Teaching experience

The course is designed for one instructor and one TA. While instructors may come from a single subdiscipline of the geosciences, the students in the course do not. To date, we have taught students geology, geophysics, atmospheric sciences, oceanography, forestry, civil environmental engineering, and biology. The typical split between undergraduates and graduates has been 50/50.

During a quarter, the course involves meeting three times a week for 90 minutes. Outside of instruction, students spend several hours (~ 5) per week on assignments, including paper reviews, homework, and their final project.

Instructors and students have access to a Jupyter Hub provisioned by 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. Additionally, unlike previous course iterations, this acceleration enabled students to complete all four pillars of the final project.

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 Jbook is designed to be a dynamic document to which the community is invited to contribute. There is much that instructors can do to bring new geoscientific data sets, produce more relevant exercises for students, improve the teaching of concepts, and keep up with ever-evolving literature.

Future improvements should include more geoscientific toy data sets, refinements of statistical learning betwen uni and multi-variate data, development of student-led exercise and additional homeworks.

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