

An Object-Oriented Implementation of Double Machine Learning using Python

Type: Editorial

Received: January 22, 2024

Published: March 20, 2024

Citation:

Rupali Taware., et al. "An Object-Oriented Implementation of Double Machine Learning using Python". PriMera Scientific Engineering 4.4 (2024): 01-02.

Copyright:

© 2024 Rupali Taware., et al. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rupali Taware^{1*}, Sunil Khilari² and Chandrani Singh³

¹Assistant Professor, Sinhgad Institute of Management, Pune, India

²Associate Professor, Sinhgad Institute of Management, Pune, India

³Director-MCA, Sinhgad Institute of Management, Pune, India

***Corresponding Author:** Rupali Taware, Assistant Professor, Sinhgad Institute of Management, Pune, India.

Introduction

Double Machine Learning (DML) is a powerful framework that combines the flexibility of machine learning with the robustness of statistical inference. It is particularly useful in settings where treatment effects are of interest, such as in econometrics and causal inference. In this article, we explore an object-oriented approach to implementing Double Machine Learning using Python, leveraging the simplicity and modularity of object-oriented programming (OOP) principles.

Python using Object-Oriented Programming

Python is a programming language that is both flexible and easy to learn. It is compatible with the object-oriented programming paradigm, which makes use of classes and objects to help organize and structure code. This method is perfect for applying complex algorithms like Double Machine Learning because it encourages modularity, reusability, and a distinct separation of responsibilities.

Designing the Double Machine Learning Class

To encapsulate the DML process, one must create a Python class that represents the DML algorithm. This class consists of various methods to handle key components of the DML framework, such as data preparation, model training, and treatment effect estimation. By organizing the code in a class, we enhance code readability, maintainability, and extensibility.

Data Preparation

The first step in any machine learning project is data preparation. Our DML class includes methods for loading and preprocessing data. This involves splitting the dataset into training and testing sets, handling missing values, and encoding categorical variables. Utilizing an object-oriented approach allows for easy customization of data preprocessing steps based on the specific requirements of the analysis.

Model Training

The DML class incorporates methods to train both the treatment model and the outcome model. The treatment model predicts the probability of receiving treatment, whereas the outcome model approximates the potential outcomes given treatment status. We employ popular machine learning libraries such as scikit-learn to implement these models within our DML class. The modular structure of the class allows users to experiment with different algorithms and hyperparameters seamlessly.

Double Machine Learning Estimation

The heart of the DML framework lies in the estimation of treatment effects. Our object-oriented implementation facilitates the computation of treatment effects using the fitted treatment and outcome models. This separation of concerns enhances code maintainability and allows users to easily swap models or modify the estimation procedure without altering the entire codebase.

Conclusion

Adopting an object-oriented approach to implement Double Machine Learning in Python brings numerous benefits, including code organization, modularity, and ease of extensibility. The resulting class encapsulates the entire DML process, providing users with a flexible and customizable tool for estimating treatment effects. This approach not only enhances code readability but also fosters collaboration and code sharing in the growing field of causal inference. As machine learning and causal inference continue to intersect, an object-oriented DML implementation in Python proves to be a valuable asset for researchers and practitioners alike.

References

1. P Bach., et al. DoubleML – An object-oriented implementation of double machine learning in R (2021): arXiv:2103.09603 [stat. ML].