The Journal of Open Source Software

ddop: A python package for data-driven operations management

Andreas Philippi[1], **Simone Buttler**[1], **and Nikolai Stein**[1]

1 Chair of Logistics and Quantitative Methods, Julius-Maximilians-Universität Würzburg, Sandering 2, Würzburg 97070, Germany

DOI: 10.21105/joss.03429

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @kramea
- @g4brielvs

Submitted: 11 May 2021
Published: 29 June 2021

DRAFT

Summary

In today's fast-paced world, companies face considerable uncertainty when making important decisions in operations management, for example, when deciding upon capacity, inventory levels, transportation, and production schedules. However, with the rise of digitization, companies have gained unprecedented access to data related to their particular decision problem, offering the opportunity to reduce the degree of uncertainty. For example, in inventory management the decision maker may have access to historical demand data as well as additional side information, such as social media data, customer behaviour, weather forecasts or calendar data. Driven by the availability of such rich data sources there has recently emerged a stream of literature in operations management research called "data-driven operations management" (DDOM). The focus of DDOM is to combine machine learning and traditional optimization techniques to prescribe cost optimal decisions directly from data. Various models have been developed and shown great performance on the dataset used. However, what is missing is efficient access to open-source code and datasets. With *ddop*, we provide a Python library that integrates well-established algorithms form the field of data-driven operations management, as well as standard benchmark datasets. Thus, *ddop* helps researchers in two ways:

- Researchers can efficiently apply and compare well-established DDOM models.
- Researchers can test new developed models on benchmark datasets provided in the package.

The application programming interface (API) of *ddop* is designed to be consistent, easy-to-use, and accessible even for non-experts. With only a few lines of code, one can build and compare various models. In *ddop* all models are offered as objects implementing the estimator interface from scikit-learn (Buitinck et al., 2013). We thus not only provide a uniform API for our models, but also ensure that they safely interact with scikit-learn pipelines, model evaluation and selection tools.

The library is distributed under the 3-Clause BSD license, encouraging its use in both academic and commercial settings. The full source code is available at https://github.com/opimwue/ddop. The package can be installed via the Python Package Index using `pip install ddop`. A detailed documentation providing all information required to work with the API can be found at https://opimwue.github.io/ddop/.

Statement of need

With the growing number of publications in the field of data-driven operations management, comparability is becoming increasingly difficult. The reasons for this are twofold: One, most scientists work with proprietary company data which cannot be shared. Two, it is not yet standard that researchers share code used to implement their models. Consequently, results are not directly reproducible and models have to be re-implemented every time a researcher wants to benchmark a new approach. This not only takes a lot of time but can also be a demanding process since such complex models are often challenging to implement. Against this background, there has recently been a call to take inspiration from the machine learning community, where great APIs like scikit-learn (Buitinck et al., 2013), fastai (Howard & Gugger, 2020), or Hugging Face (Wolf et al., 2019) have been developed that allow previous developed ML models to be effectively applied on different dataset. Following up on this, *ddop* is the first of its kind to integrate well-established data-driven models for operations management tasks. At the current state, this includes various approaches to solve the data-driven newsvendor problem, such as weighted sample average approximation (Bertsimas & Kallus, 2020), empirical risk minimization (Ban & Rudin, 2019), and a deep learning based approach (Oroojlooyjadid et al., 2020). In addition, the library provides different real-world datasets that can be used to quickly illustrate the behaviour of the available models or as a benchmark for testing new models. *ddop's* aim is to make data-driven operations management accessible and reproducible.

Usage

Since all models in *ddop* implement the estimator interface from *scikit-learn* consisting of a *fit*, *predict*, and *score* method, usage follows the standard procedure of an *scikit-learn* regressor. First, a model is initialized by calling the class constructor from a given set of constant hyper-parameter values, each describing the model or the optimisation problem the estimator tries to solve. Note that for ease of use, all estimators use reasonable default values. It is therefore not necessary to pass any parameter to the constructor. However, it is recommended to tune them for the respective application, since this can often improve decision quality. After the model has been initialized, the *fit* method is used to learn a decision model from the training data (*X_train*, *y_train*). Once the training process is completed, the function returns the fitted model, which can then be used to make decisions for new data (*X_test*) by using the *predict* method. Finally, the score method can be used to access the decision quality of a model. The method takes as input *X_test* as well as the corresponding true values *y_test* and computes the average costs between *y_test* and *predict(X_test)*. Because all estimators follow the same interface, using a different model is as simple as replacing the constructor.

Future Work

There are several directions that the ddop project aims to focus on in future development. While at the current state there are only algorithms available to solve the newsvendor problem, the goal is to include models to solve other operations management task like multi-period inventory management or capacity management. In addition, we aim to extend the library in terms of available datasets and tutorials.

References

Ban, G.-Y., & Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, *67*(1), 90–108. https://doi.org/10.1287/opre.2018.1757

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, *66*(3), 1025–1044. https://doi.org/10.1287/mnsc.2018.3253

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., & others. (2013). API design for machine learning software: Experiences from the scikit-learn project. *arXiv Preprint arXiv:1309.0238*.

Howard, J., & Gugger, S. (2020). Fastai: A layered API for deep learning. *Information*, *11*(2), 108. https://doi.org/10.3390/info11020108

Oroojlooyjadid, A., Snyder, L. V., & Takáč, M. (2020). Applying deep learning to the newsvendor problem. *IISE Transactions*, *52*(4), 444–463. https://doi.org/10.1080/24725854.2019.1632502

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & others. (2019). HuggingFace's transformers: State-of-the-art natural language processing. *arXiv Preprint arXiv:1910.03771*.