Software Impacts 12 (2022) 100240

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Data Science Toolkit: An all-in-one python library to help researchers and practitioners in implementing data science-related algorithms with less effort

Chouaib El Hachimi [a,*], Salwa Belaqziz [a,b], Saïd Khabba [a,c], Abdelghani Chehbouni [a,d]

[a] Mohammed VI Polytechnic University (UM6P), Center for Remote Sensing Applications (CRSA), Benguerir, Morocco
[b] UIZ University, Faculty of Science, LabSIV Laboratory, Department of Computer Science, Agadir, Morocco
[c] Cadi Ayyad University, Faculty of Sciences Semlalia, LMFE, Department of Physics, Marrakesh, Morocco
[d] Université de Toulouse, Centre d'Etudes Spatiales de la Biosphère (CESBIO), Toulouse, France

ARTICLE INFO

Keywords:
Data science
Machine learning
Data processing
Data visualization
Data representation

ABSTRACT

Data Science Toolkit (DST) is a python library built as a wrapper layer on top of several libraries to increase the abstraction level of the code, making its users more efficient and productive. The current version is widely used in our ongoing research activities that focus on optimizing agricultural management practices using artificial intelligence. DST adopts an object-oriented approach in implementing data science algorithms and is therefore composed of multiple classes such as the DataFrame class that adds additional functionalities to the standard pandas dataframe and the Model class that facilitates the building, training, and evaluation of machine learning models.

Code metadata

Current code version	*v0.0.1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-182
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/3902531/tree/v1
Legal Code License	*MIT license (MIT)*
Code versioning system used	*git*
Software code languages, tools, and services used	*Python*
Compilation requirements, operating environments & dependencies	pandas, keras, nltk, scikit-learn, wordcloud, tensorflow, scipy, numpy, matplotlib, seaborn, plotly, geopandas, openCV, XGboost
If available Link to developer documentation/manual	https://data-science-toolkit.readthedocs.io
Support email for questions	elhachimi.ch@gmail.com; salwa.belaqziz@gmail.com

1. Introduction

Data is considered the oil of the 21st century, and processing it is therefore essential for success. Data can be used in almost all domains, including providing quantitative guidance on future business strategies and operations [1], helping diagnose disease in healthcare [2,3], supporting decision-making for policymakers [4], monitoring the Earth for sustainability [5], preventing severe climate events, analyzing the big data generated by sensors to make agriculture more sustainable [6], and to participate in resolving global hunger challenges.

Data science emerges as the new independent knowledge domain that provides the necessary skills to deal with, and conduct projects related to this new resource (data). A successful data scientist must have several prerequisites such as mathematics (statistics, linear algebra, graph analysis, optimization, etc.), computer science (databases, storage, visualization, programming, etc.), and domain knowledge of the problem at hand.

Various programming languages such as R and MATLAB are used by data scientists to perform tasks like statistics, matrix calculations,

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: chouaib.elhachimi@um6p.ma (C. El Hachimi).

https://doi.org/10.1016/j.simpa.2022.100240
Received 10 December 2021; Received in revised form 2 January 2022; Accepted 20 January 2022

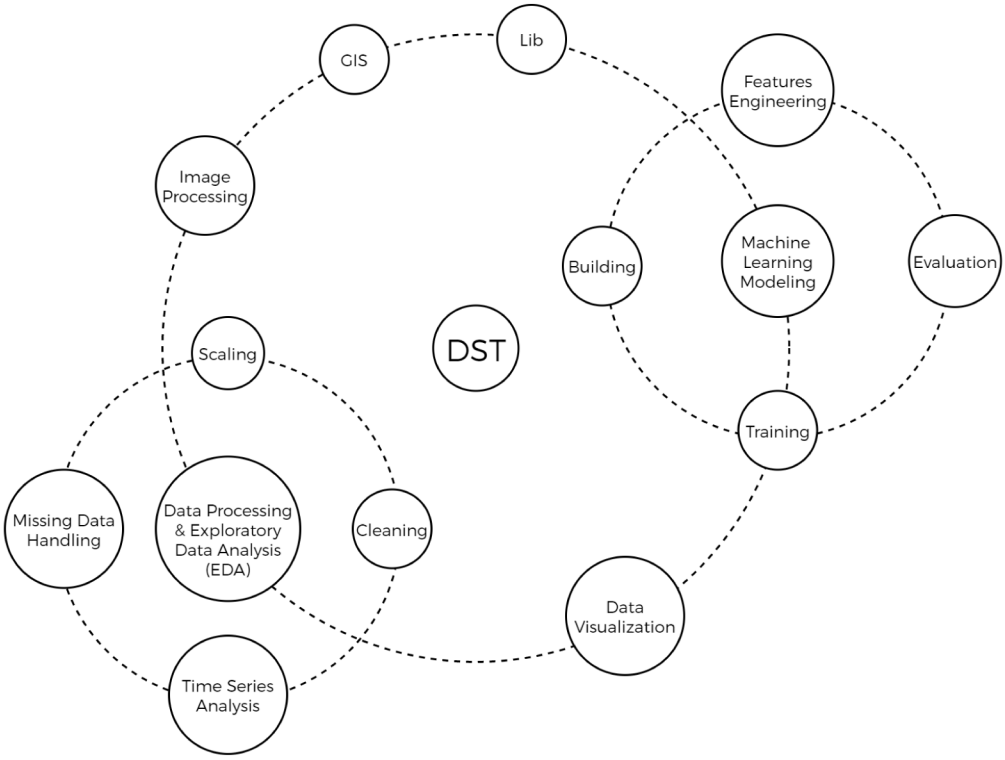

Fig. 1. The general architecture of DST.

visualization, and so on. However, Python has become the most popular tool to use in the data science field because it is easy to learn no matter what your background or experience is, and has a huge community to help in case of errors in almost all domains. The diversity and availability of a huge number of libraries is also an important contributing factor to its celebrity, making it the best choice for a data scientist. For example, the mathematical library NumPy [7] for scientific computing, Pandas [8] for data analysis, Skit-learn [9] and TensorFlow [10] for machine learning and deep learning applications, not to mention that there are also Python packages that allow calling code in other languages to benefit from the speed of compiled code such as the case for NumPy. Additionally, Python is suitable for both development and production environments.

To contribute from our perspective to the Python community, we developed the Data Science Toolkit library (DST) which is an open-source, low-code library that comes as a wrapper on top of various commonly used libraries and comes with additional functionalities with the goal of using as little code as possible and keeping the code closer to the human language.

2. Functionalities

The Data Science Toolkit Library adopts an Object-Oriented approach to encapsulate and hide information, enabling its users to be more productive and focus on the problem instead of dealing with the level of detail and internal implementations that do not matter to them. The current version contains six classes (Fig. 1), where each class contains a collection of data (attributes) with associated behaviors (methods or functions) and can interact with and serve other classes when needed.

2.1. Data processing and exploratory data analysis

The DST provides various data processing and exploratory analysis algorithms that enhance the standard data structure used by the pandas's library. These functionalities include, but are not limited to:

- Dealing with missing data by automatically providing statistics of them, and then apply different implemented filling methods.
- Various column transformation and features engineering methods such as one-hot encoding, scaling using the MinMax, Standard or custom scalers, or transforming a column by applying calculations using a combination of other columns, etc.
- Querying, filtering and searching in a particular column either by using regular expressions patterns or by using SQL-like expressions.
- Dealing with time series by providing data imputation methods, performing analyses, downscaling, upscaling, and transforming them into data generators that are an example of the accepted input for supervised machine learning regression models in forecasting and prediction tasks.

2.2. Data visualization

Human is a visual creature, and he cannot interpret or get insights from large amounts of data. By using visualizations, he gets a better idea of what the data at hand contains in terms of patterns or correlations that may exist in it. In data science, data visualization is a crucial step, and the Chart class is the DST response to this. It accepts a dataframe object as a parameter and then creates data visualizations depending on requirements. It is built on top of mathplotlib [11], seaborn [12], and other libraries to provide various types of charts such as comparison plots, relationship plots, composition plots, distribution plots, and geoplots.

2.3. Machine learning modeling

The DST contains the class Model that is designed to make machine learning modeling straightforward. Before training models, DST allows feature selection that is a vital step in improving the models' performance. This is done by calculating the features' importance or relevance by measuring the predictive impact of each feature. Additionally, DST enables building various machine learning models (K-Nearest

Neighbors [13], Decision Tree [14], Random Forest [15], Naive Bayes, Support Vector Machine [16], Linear and Logistic Regression, XQ-boost [17]) and deep learning models [18] (Feed Forward Neural Network, Convolutional Neural Network, Recurrent Neural Network, Long Short Term Memory [19]) with ease and in just a few lines of code. It then does the data splitting and shuffling it randomly internally and provides the most used evaluation metrics for both classification (confusion matrix, accuracy, recall, precision, f-score, Receiver Operating Characteristic, Area Under the Curve) and regression (The coefficient of determination R^2, Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, Median Absolute Error, and Mean Squared Log Error) tasks after the training is completed. It also manages the import and export of trained or pre-trained models for deployment in a production environment.

2.4. Lib class of DST

The Lib class of DST is a static class that serves other components with implementations of auxiliary algorithms with the objective of not reinventing the wheel. These algorithms range from mathematical functions (The Greatest Common Divisor, the Least Common Multiple, verifying a prime number, decomposition into prime numbers, etc.), to read/write files, string manipulation, etc. In addition, users can add their own customized functions to this class, which may help them in implementing methods depending on their specific needs and requirements, which is true for other classes as well since DST was developed with reusability, clean code (readable, simple, and concise) and modularity in mind.

2.5. Image processing

DST has an ImageFactory class that deals with Image I/O and display operations. This class includes implementations of image processing algorithms such as resizing, cropping, rotating, calculating and plotting histograms, grayscale conversion, thresholding or binarization, applying convolutions with different types of filters or masks, finding contours, and between-image mathematical equations, and so on.

2.6. Geographic information system

The GIS class provides several functions to facilitate the processing of geospatial data (shapefiles, geoJSON, etc.) and stores this data as layers. It enables visualizing these layers in a customizable geographical map, calculating distances and areas, data transforming, adding/editing new shapes, and more.

3. Impact

The Data Science Toolkit library is under active and continuous development. The current version has been used already in our research activities, among others, to: (1) crop recommendation and weather forecasting [20], where we investigated several machine learning and deep learning models to recommend the best crop to grow and forecast the hourly average air temperature using the Long-Strong-Term Memory (LSTM) that represents the next generation of Recurrent Neural Network (RNN) and Facebook prophet model [21]. (2) The accepted paper titled "Early estimation of daily reference evapotranspiration using machine learning techniques for efficient management of irrigation water" where DST was used in all phases of implementing the proposed method, including cleaning the meteorological data, data analysis and handling missing values, calculation of correlations between different meteorological data, measuring the importance of wind speed, wind direction, relative humidity, global solar radiation, air temperature, and rainfall for the prediction of the target variable which is the reference evapotranspiration (ET_0). DST was also used in the machine learning modeling phase ranging from building and training, to the evaluation of the Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Linear Regression and XGBoost machine learning models; not to mention ongoing research work where DST is used in our day-to-day development, implementation, and testing of hypotheses and proposed research methods related to using artificial intelligence to determine the itinerary for optimal crop growth and development with efficient management of irrigation water.

4. Further development

Regarding our future roadmap, we will certainly continue using the library in our future research work and encourage other researchers to use it by promoting it in scientific events and developing simplified documentation and tutorial notebooks that cover all the power it provides, especially for educational purposes. We are also aiming to expand the Data Science Toolkit by adding more classes and implementations related to the data science field, such as reinforcement learning, unsupervised learning, and heuristic algorithms, to name a few. In addition, we will continue to enhance and improve the current classes to keep the library up to date. Not to mention that the open-source community can also contribute to the library by sending GitHub Pull Requests.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This study was supported by and conducted within the Center for Remote Sensing Applications (CRSA) (https://crsa.um6p.ma/), at the Mohammed VI Polytechnic University (UM6P) in Morocco.

References

[1] Mauriciusa Munhoz de Medeiros, Norberto Hoppen, Antonio Carlosa Gastaud Maçada, Data science for business: benefits, challenges and opportunities, Bottom Line (ISSN: 0888045X) 33 (2020) 149–163, http://dx.doi.org/10.1108/BL-12-2019-0132/FULL/XML.

[2] Habib Dhahri, et al., Automated breast cancer diagnosis based on machine learning algorithms, J. Healthc. Eng. (ISSN: 20402309) 2019 (2019) http://dx.doi.org/10.1155/2019/4253641.

[3] Chouaiba El Hachimi, Abdessadek Aaroud, Medical use of deep learning: Malaria testing using pre-trained ResNet, in: Mostafa Ezziyyani (Ed.), Advanced Intelligent Systems for Sustainable Development, AI2SD'2019, Springer International Publishing, Cham, 2016, pp. 273–280, http://dx.doi.org/10.1007/978-3-030-36664-3_31, ISBN: 978-3-030-36664-3.

[4] Nada Elgendy, Ahmed Elragal, Big data analytics in support of the decision making process, Procedia Comput. Sci. (ISSN: 1877-0509) 100 (2016) 1071–1084, http://dx.doi.org/10.1016/J.PROCS.2016.09.251.

[5] Gilberto Camara, et al., Big earth observation data analytics: Matching requirements to system architectures, in: Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial 2016, Oct. 2016, pp. 1–6, http://dx.doi.org/10.1145/3006386.3006393.

[6] Sjaak Wolfert, et al., Big data in smart farming – A review, Agric. Syst. (ISSN: 0308-521X) 153 (2017) 69–80, http://dx.doi.org/10.1016/J.AGSY.2017.01.023.

[7] Stéfan Van Der Walt, S. Chris Colbert, Gaël Varoquaux, The NumPy array: A structure for efficient numerical computation, Comput. Sci. Eng. (ISSN: 15219615) 13 (2) (2011) 22–30, http://dx.doi.org/10.1109/MCSE.2011.37, arXiv:1102.1523.

[8] Wes Mckinney, pandas: a foundational python library for data analysis and statistics, URL: http://pandas.sf.net.

[9] Fabiana Pedregosa FABIANPEDREGO.S.A., et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. (ISSN: 1533-7928) 12 (85) (2011) 2825–2830, http://jmlr.org/papers/v12/pedregosa11a.html.

[10] Martin Abadi, et al., Tensorflow: A system for large-scale machine learning, 2016, https://research.google/pubs/pub45381/.

[11] P. Barrett, et al., Matplotlib – a portable python plotting package, in: ASPC, Vol. 347, 2005, p. 91, https://ui.adsabs.harvard.edu/abs/2005ASPC..347...91B/abstract.

[12] Michaela L. Waskom, Seaborn: statistical data visualization, J. Open Source Softw. (ISSN: 2475-9066) 6 (60) (2021) 3021, http://dx.doi.org/10.21105/JOSS.03021, https://joss.theoj.org/papers/10.21105/joss.03021.

[13] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Statistician 46 (3) (1992) 175–185, http://dx.doi.org/10.1080/00031305.1992.10475879.

[14] J.R. Quinlan, Decision trees and decisionmaking, IEEE Trans. Syst. Man Cybern. 20 (2) (1990) 339–346, http://dx.doi.org/10.1109/21.52545.

[15] Tina Kam Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844, http://dx.doi.org/10.1109/34.709601.

[16] Corinna Cortes, Vladimir Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, http://dx.doi.org/10.1023/A:1022627411411.

[17] Tianqi Chen, Carlos Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-August-2016, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[18] Yann Lecun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 2015 521:7553 (ISSN: 1476-4687) (2015) 436–444, http://dx.doi.org/10.1038/nature14539, https://www.nature.com/articles/nature14539.

[19] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. (ISSN: 08997667) 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[20] Chouaib El Hachimi, et al., Towards precision agriculture in Morocco: A machine learning approach for recommending crops and forecasting weather, in: 2021 International Conference on Digital Age Technological Advances for Sustainable Development, ICDATA, 2021, pp. 88–95, http://dx.doi.org/10.1109/ICDATA52997.2021.00026.

[21] Sean J. Taylor, Benjamin Letham, Forecasting at scale, The American Statistician (ISSN: 15372731) 72 (1) (2018) 37–45, http://dx.doi.org/10.1080/00031305.2017.1380080.