Software Impacts 14 (2022) 100354

Contents lists available at ScienceDirect

Software Impacts

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

Original software publication

An eXplainable Artificial Intelligence tool for statistical arbitrage

Salvatore Carta [a], Sergio Consoli [b,*], Alessandro Sebastian Podda [a], Diego Reforgiato Recupero [a], Maria Madalina Stanciu [a]

[a] *Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari (CA), Italy*
[b] *European Commission, Joint Research Centre (JRC), Via E. Fermi 2749, I-21027 Ispra (VA), Italy*

ARTICLE INFO

Keywords:
Machine learning
Quantitative finance
Time-series forecasting
eXplainable Artificial Intelligence
Stock market forecast
Statistical arbitrage

ABSTRACT

This open-source tool, written in Python, referred to as *XAI StatArb*, implements a machine learning approach (ML) powered by eXplainable Artificial Intelligence techniques integrated into a statistical arbitrage trading pipeline. Specifically, given a set of stocks and their raw financial information, the tool aims at forecasting the next day's return. Based on the predicted return, we trade the underperforming and overperforming stocks. Additionally, the tool contains three ML methods to discard irrelevant features for the prediction task. They are aimed at improving not only the prediction performance at the stock level but also overall at the stock set level.

Code metadata

Current code version	v2.1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-110
Permanent link to reproducible capsule	https://codeocean.com/capsule/8369112/tree/v2
Legal Code License	MIT License: https://opensource.org/licenses/MIT
Code versioning system used	Git/GitHub
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python ≥ 3.8.5, NumPy ≥ 1.19.2, SciPy ≥ 1.5.2, Scikit-learn ≥ 0.23.2, Pandas ≥ 1.2.1, Matplotlib ≥ 3.3.4, Lime ≥ 0.2.0.1
If available Link to developer documentation/manual	https://github.com/Artificial-Intelligence-Big-Data-Lab/xai_statarb/blob/main/README.md
Support email for questions	madalina.stanciu@unica.it

1. Introduction

Machine learning (ML) techniques have recently become the norm for detecting patterns in financial markets [1,2]. However, relying solely on ML algorithms for decision-making can have negative consequences, especially in a critical domain such as the financial one. On the other hand, it is well-known that transforming data into actionable insights can pose a challenge even for seasoned practitioners, particularly in the financial world. Given these compelling reasons, this work, whose implemented code is available in [3], proposes an ML approach powered by eXplainable Artificial Intelligence techniques integrated into a statistical arbitrage trading pipeline (a.k.a. *XAI StatArb*). Specifically, given a set of stocks and their raw financial information we aim at forecasting the next day's return. Based on the predicted return we trade the underperforming and overperforming stocks. Additionally, we propose three methods to discard irrelevant features for the prediction task.

For this purpose, we provide a code that:

DOI of original article: https://doi.org/10.1016/j.eswa.2022.117763.

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 addresses: salvatore@unica.it (S. Carta), sergio.consoli@ec.europa.eu (S. Consoli), sebastianpodda@unica.it (A.S. Podda), diego.reforgiato@unica.it (D. Reforgiato Recupero), madalina.stanciu@unica.it (M.M. Stanciu).

https://doi.org/10.1016/j.simpa.2022.100354
Received 28 June 2022; Accepted 1 July 2022

1. manages the setting of the algorithm parameters, implements the data acquisition, and trains the model;
2. implements the features selection process,
3. and provides a framework for the execution of the statistical arbitrage strategy.

Given a study period, the algorithm performs the following steps. First, for each stock, the related financial data are collected, and the algorithm extracts the input features and the target variable. The employed input features are the price returns, a series of technical indicators, or a combination of the two. As for the target variable, we use the next-day return.

Second, using an ML model trained with the whole feature set, the algorithm computes the feature importance score. Based on them, features are divided into two categories: important features and unimportant ones. The unimportant features are then discarded, and a new model is trained. The new model receives as input only the features that are considered to be important and makes consequent predictions. The loss of the newly trained model is subtracted from the loss of the base regressor. The loss difference is used to compute an optimal feature score threshold, below which features are indeed considered unimportant for the entire stock set. The described process constitutes the learning phase and is performed on *In-Sample* data.

Third, the algorithm performs the forecasting and trading on *Out-of-Sample* data, i.e. data "unseen" by the models in the processes of feature selection of model training. In the trading phase for each stock, only the model trained with the optimal features set is used.

For strategy backtesting, we use the *walk-forward* validation method [4], which is a common approach for validating time-series data in finance. It consists of splitting the study period of the time-series data into overlapping training periods, also known as *In-Sample*, and non-overlapping test (trading) periods, also referred to as *Out-of-Sample*. Each tuple *In-Sample* and *Out-of-Sample* data constitutes a walk. The *In-Sample* data is further split into two chunks: *training*, used for ML model training, and *validation*, used for relevant feature selection at the entire stock set.

2. Functionalities and key features

The software was implemented using the Python programming language [5], relying on several packages, most importantly: *scikit-learn* [6], *numpy* [7], and *pandas*[1] [8]. The source code is divided into several classes and modules. The most important are:

- `main.py` : This represents the main entry point of the framework where the whole workflow is defined. It can run experiments from the command line, as well as generate experimental setups and results.
- `CompanyFeatures`: This class is used to process the raw financial data, and to produce, as output, the features, lagged returns or technical indicators, the target variable, and the next day's return.
- `Threshold`: This class computes the optimal threshold below which features should be removed. In other words, it computes the features to be removed at the entire stock set.
- `StatArbRegression`: This class contains the essential methods for performing Statistical Arbitrage and provides the financial results per walk of the trading, as well as the daily traded companies.

3. Impact overview

The *XAI StatArb* framework is primarily and specifically used for financial forecasting and trading using a Statistical Arbitrage strategy. While there are several tools for trading and evaluating a trading strategy, to the best of our knowledge there is no framework designed for combining machine learning, feature selection, and Statistical Arbitrage in a unified framework. The development of this software allowed the pursuit of two of our existing research questions: (1) How can we select the appropriate input features for an ML-driven trading system? (2) Are the same input features relevant to other related stocks? For these purposes, we needed to perform experiments using the software to test any developed approaches in a real setting, as reported in the following scientific publications:

1. Salvatore Carta, Sebastian Podda, Diego Reforgiato Recupero, and Maria Madalina Stanciu. Explainable AI for financial forecasting. In The 7th International Conference on Machine Learning, Optimization, and Data Science (LOD 2021). Lecture Notes in Computer Science, 2022, pp. 51–69. See [9]
2. Salvatore Carta, Sebastian Podda, Diego Reforgiato Recupero, and Maria Madalina Stanciu. Statistical arbitrage powered by Explainable Artificial Intelligence. Expert Systems with Applications 206 (2022) 117763. See [10]

We evaluated the approaches on historical data of component stocks of the S&P500 index and aimed at improving not only the prediction performance at the stock level but also overall at the stock set level. Using this software, in these publications we were able to show improvements upon simple baselines when applied to real-world financial data.

Additionally, one of the strengths of the proposed framework is that it is entirely written in Python, a programming language with an easy-to-understand syntax that has been largely used in several recent ML-related development works. This allows for effortless reuse of already implemented routines or an easy extension with new functionalities. Additionally, Python's flexibility facilitates the integration with various libraries for evaluating any trading strategy, and additional modules coming also from other programming languages.

The *XAI StatArb* algorithm has been involved in several industrial collaborations and projects, matching the demand for an intuitive, easily integrable tool for statistical arbitrage trading. It was exploited during the collaboration with the Joint Research Centre of the European Commission to develop a procedure for large scale identification of relevant economic events in Europe from news and social media.

4. Conclusions and further development

The *XAI StatArb* package provides an easy-to-use framework to integrate eXplainable Artificial Intelligence techniques into a statistical arbitrage trading pipeline for financial forecasting. Researchers and developers can take advantage of this package due to its simplicity.

At the current stage, we are aware of several points of possible improvement for the released software. For example, we only considered a limited type of input features, i.e., lagged returns and technical indicators, while others could also be included. Furthermore, the number of input features is currently needed to be statically fixed a-priori, while a dynamic setting approach would be more flexible to adapt to different real-world scenarios. Additionally, determining feature importance is only based on permutation, while other feature importance methods could be incorporated into the framework as well.

[1] pandas-dev/pandas: Pandas 1.4.2, available at https://doi.org/10.5281/zenodo.3509134.

CRediT authorship contribution statement

Salvatore Carta: Visualization, Investigation, Formal analysis, Writing – original draft, Validation, Supervision, Project administration, Writing – review & editing. **Sergio Consoli:** Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Investigation, Writing – review & editing. **Alessandro Sebastian Podda:** Software, Data curation, Formal analysis, Writing – original draft, Visualization, Investigation, Writing – review & editing. **Diego Reforgiato Recupero:** Visualization, Investigation, Formal analysis, Writing – original draft, Validation, Supervision, Project administration, Writing – review & editing. **Maria Madalina Stanciu:** Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Investigation, Writing – review & editing.

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.

Acknowledgments

The authors would like to acknowledge the support of the Joint Research Centre of the European Commission. The views expressed are purely those of the authors and may not in any circumstance be regarded as stating an official position of the European Commission.

References

[1] S.M. Carta, S. Consoli, L. Piras, A.S. Podda, D. Reforgiato Recupero, Explainable machine learning exploiting news and domain-specific lexicon for stock market forecasting, IEEE Access 9 (2021) 30193–30205, http://dx.doi.org/10.1109/ACCESS.2021.3059960.

[2] S. Carta, A. Corriga, A. Ferreira, A.S. Podda, D. Reforgiato Recupero, A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning, Appl. Intell. 51 (2) (2021) 889–905, http://dx.doi.org/10.1007/s10489-020-01839-5.

[3] M.M. Stanciu, D. Reforgiato Recupero, S. Podda, S. Carta, S. Consoli, Statistical Arbitrage Powered by Explainable Artificial Intelligence [Source Code], Code Ocean, 2022, http://dx.doi.org/10.24433/CO.2649328.v2.

[4] S.M. Carta, S. Consoli, A.S. Podda, D. Reforgiato Recupero, M.M. Stanciu, Ensembling and dynamic asset selection for risk-controlled statistical arbitrage, IEEE Access 9 (2021) 29942–29959, http://dx.doi.org/10.1109/ACCESS.2021.3059187.

[5] G. Van Rossum, F.L. Drake Jr., Python reference manual, 1994.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[7] C.R. Harris, K.J. Millman, S.J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, et al., Array programming with NumPy, Nature 585 (7825) (2020) 357–362.

[8] W. McKinney, Data structures for statistical computing in python, in: S. van der Walt, J. Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61, http://dx.doi.org/10.25080/Majora-92bf1922-00a.

[9] S. Carta, S. Podda, D. Reforgiato Recupero, M.M. Stanciu, Explainable AI for financial forecasting, in: The 7th International Conference on Machine Learning, Optimization, and Data Science, Vol. 13164, LOD 2021, Lecture Notes in Computer Science, 2022, pp. 51–69, http://dx.doi.org/10.1007/978-3-030-95470-3_5.

[10] S. Carta, S. Consoli, A.S. Podda, D. Reforgiato Recupero, M.M. Stanciu, Statistical arbitrage powered by explainable artificial intelligence, Expert Syst. Appl. 206 (2022) 117763, http://dx.doi.org/10.1016/j.eswa.2022.117763.