# Strijk Philippe - BP voorstel - Machine Learning pipeline for trading bots using PySpark

Strijk Philippe [1]

**Samenvatting**
A general outline of how to approach automated trading bots using machine learning. Includes a short overview of existing callenges in making a machine learning model that can lucratively optimize portfolios and make trades.

**Sleutelwoorden**
Apache — Spark — Big Data — Machine Learning — Trading — Economics — Reinforcement Learning

**Co-promotor**
[2] (Manu Lahariya)

**Contact:** [1] philippe.strijk@student.hogent.be [2] manu me.lahariya.001@gmail.com

## Inhoudsopgave

## 1. Introduction

With the rise of crypto and general digitalization of currencies, it is a no-brainer that trading on the stock market will be a place where a regular human will get outperformed by sophisticated AI systems. The marketplace will be one where AI's will battle each other for the largest profit margins. (Ianenko e.a., 2019) This paper will discuss the possibility of using Apache Spark and machine learning techniques to help achieve automated trading.

The challenges along the way include, but are not limited to:

- Data quality and availability
- Model selection and optimization
- Overfitting
- Implementation and deployment
- Risk management

The main research question is as follows: Is it feasible to create a Machine Learning pipeline for automated trading using Spark in Python?

The results can be verified if the system is:

- Scalable and able to proess large amounts of data very quickly
- Efficient computation
- Ready for integration into an existing big data ecosystem.

If a general implementation of a machine learning pipeline for automated trading bot is possible, it should be considered as a success. If the automated bot is capable of predicting a net positive portfolio gain, it should be considered a huge success.

## 2. State-of-the-art

Automated trading systems are programs that use algorithms and machine learning to trade financial instruments automatically. These bots make decisions on when to buy and sell assets. Financial instruments that can be used to trade are stocks, bonds, mutual funds, futures, options, commodities, currencies, and derivatives. The state-of-the-art in trading bots is constantly evolving, as new techniques and technologies are developed.

Some current trends include the use of natural language processing (NLP) and machine learning techniques to analyze and interpret financial news and social media data, and the use of high-frequency trading (HFT) algorithms to execute trades at high speeds. There is also increasing interest in the use of artificial intelligence (AI) and deep learning techniques to develop more sophisticated and adaptive trading bots.

In general, trading bots are becoming more advanced and sophisticated, with the ability to analyze and interpret a wide range of data sources and make increasingly sophisticated trading decisions.(Ferreira e.a., 2021) However, trading bots also face significant regulatory and ethical challenges, and there is ongoing debate about the role and impact of automated trading systems in financial markets.

The main discussion point remains: If AI is omnipresent in the market, what will be the impact on economic growth? Will it hurt small and medium sized businesses?

## 3. Methodology

At first, a sober analysis has to be made as for the scope of the project. What can be done in the amount of time given to write this paper? What are the necessary steps? Should sprints be implemented? When should results be produced? When is progress shown?

To be able to have a reference for Spark performance, the entire process will be compared to tensorflow libraries.

Not unimportant: what is the timeframe for the entire process? The timeline below is a rough estimate for how long each task should take.

The possible pipeline for making an automated bot include (Zhang e.a., 2022):

- Getting high quality data: week 1
- Feature engineering and data cleaning: week 2
- Picking the right model(s), tuning the parameters and cross-validating: week 3 & 4
- Predictions that include: risk factors, prices & returns
- Asset selection: week 5
- Portfolio optimizer: week 6
- Optimizing trade on selected portfolio: week 7
- Making the trade calls in real time: week 8 & 9
- Monitoring and evaluating the model: what is the performance? How can the system be better?: week 10
- Self-optimization: week 11 & 12

The outline above will be programmed in Python, using Apache Spark libraries whenever possible.

There is plenty of stock-market trading data available on the web. At first, to set up the pipeline, historic data will be used. The selection process of the dataset will be documented. By the end, hopefully, the system will be able to pick up on real-time data.

After data selection, the feature selection takes place. This part of the research will focus heavily on the economic aspect. How does the data look that is passed on to existing models?

Picking the model will be possible after the data is selected. Supervised learning algorithms are well-suited, because they can accurately predict outcomes based on a set of input variables. Unsupervised learning algorithms, such as clustering, can be used to identify patterns in market data and identify opportunities. Deep learning algorithms can be used to recognize patterns in large datasets. Alternatively, reinforcement learning in Spark should be used to maximize returns and minimize risks. Azhikodan e.a. (2018) Huang e.a. (2022)

By asset selection is meant that the model should be able to reduce risk by diversifying into different sectors. Should this be rule-based or model-based? What stocks should be chosen? Should there be a different classification model for asset selection?

How should the portfolio be optimized? What weights are set per sector and what is the risk-return profile? Again, rule-based or model-based?

Next step is to apply the model and evaluate the precision. Has revenue been made? Does Spark work correctly? Are there ways to reduce the data without losing performance?

Finally, can the system self-optimize based on real-time data?

## 4. Expected results, conclusion

As mentioned before, the conclusions of this paper will be made by comparing the running-time and accuracy of the model between Spark and tensorflow.

The results are satisfactory if: 1) the system outperforms the average return on global index funds, which was 8.81% in 2020 and 2) The Spark system outperforms Tensorflow.

## Referenties

Azhikodan, A. R., Bhat, A. G. K., & Jadhav, M. V. (2018). Stock Trading Bot Using Deep Reinforcement Learning, 41–49. https://doi.org/10.1007/978-981-10-8201-6_5

Ferreira, F. G. D. C., Gandomi, A. H., & Cardoso, R. T. N. (2021). Artificial Intelligence Applied to Stock Market Trading: A Review. *IEEE Access*, *9*, 30898–30917. https://doi.org/10.1109/access.2021.3058133

Huang, X., Zhang, H., & Zhai, X. (2022). A Novel Reinforcement Learning Approach for Spark Configuration Parameter Optimization. *Sensors*, *22*(15), 5930. https://doi.org/10.3390/s22155930

Ianenko, M., Ianenko, M., Huhlaev, D., & Martynenko, O. (2019). Digital transformation of trade: problems and prospects of marketing activities. *IOP Conference Series: Materials Science and Engineering*, *497*, 012118. https://doi.org/10.1088/1757-899x/497/1/012118

Zhang, L., Wu, T., Lahrichi, S., Salas-Flores, C.-G., & Li, J. (2022). A Data Science Pipeline for Algorithmic Trading: A Comparative Study of Applications for Finance and Cryptoeconomics. *The First International Symposium on Recent Advances of Blockchain Evolution: Architecture, Intelligence, Incentive, and Applications*. https://doi.org/10.48550/ARXIV.2206.14932