Trality Wizard Challenge

Davide Scassola

Methodology

First of all, I set up the whole architecture, programming the three classes Pipeline, MyModel and MyPreprocessor according to the interface defined by the main class, making almost no adjustments to this last one. According to my implementation, the class MyPreprocessor handles generic data processing tasks like transforming a time series into the series of log-returns or trivially, loading the dataset into memory. Instead, other model-specific pre-processing steps (like feature extraction) are implemented in the model class. So, I first implemented a “dummy” model, in order to check the correctness of the pipeline, then I started thinking about an effective model.

Model

As often done when analyzing stocks time series, that can be modeled as random walks, I transformed each series into the series of log-returns, then I normalized the dataset (global mean and average). Therefore, I mainly tested three models: a linear auto-regressive model, a simple recurrent neural network (based on stacked LSTMs), and xgboost. No one of these three models could consistently improve the accuracy of the “bias” model of more than 1.5%, so I decided to implement xgboost (with a bit of preliminar features extraction) since at least the training is relatively fast with respect to a RNN.

Conclusion

The results in terms of performances are quite disappointing (about 1% of accuracy improvement), anyway, the models I tried are not the most sophisticated I could think of, but are usually a good compromise between effectiveness and ease of implementation (in terms of time).

Unfortunately I couldn’t achieve the wizard’s accuracy, I hope that at least the code is sound and reusable.