Two Sigma Using News to Predict Stock Movements

Model Summary Documentation

Andrzej Urbanowicz	Krzysztof Urbanowicz
andrzej.urbanowicz@gmail.com	wonabru@gmail.com

Table of contents

A1. Team information

Competition Name: **Two Sigma Using News to Predict Stock Movements**

- Team Name: **U Technology**
- Private Leaderboard Score: **0.79583**
- Private Leaderboard Place: **2**

Team:

Andrzej Urbanowicz Warsaw, Poland/Los Angeles, USA andrzej.urbanowicz@gmail.com	**Krzysztof Urbanowicz** Gdynia, Poland wonabru@gmail.com

A2. Team background

Andrzej Urbanowicz	MSc in Computer Science Bioinformatics [RNA Sequencing analysis at the University of California Los Angeles (UCLA), USA] Big Data Architect/Data Engineer/Software Engineer working in Poland (Warsaw), Switzerland, and Los Angeles (USA) 15+ years experience in programming while working for big international IT companies such as IBM, Hewlett Packard (HP) and CERN and startups.

Krzysztof Urbanowicz	PhD in Physics Post-Doc in Max Planck Institute in Dresden Many years working as a Quant specialist, especially experienced in derivatives (futures, options) Experience in nonlinear time series analysis and econophysics 15+ years experience in finance and algo trading

- **What is your academic/professional background?**

  Andrzej holds Msc in Computer Science on optimization specialization. Hi did master's thesis in telecommunication network optimization. He also did his postgraduate study in project management at the Warsaw School of Economics (SGH). He has 15+ years professional experience in computer programing including Java, C++, R, Python, etc. He used to work for CERN, Hewlett Packard (HP) and IBM as a Software Developer and later Architect. He moved to the USA 4 years ago. He has been working as Big Data Architect in two start-ups. He is also involved in multiple bioinformatics projects at the University of California at Los Angeles (UCLA) bearing on RNA Sequencing and biomarker development for human systemic diseases such as cancer. He did several courses related to Data Science.

  Krzysztof has a PhD in physics finished in 2004. He works mostly on the applications of physics in financial analysis. He is a specialist in the chaos theory, econophysics, and time series analysis. He managed to develop some new methodologies on noise level estimations and reduction from time series, etc. He published 19 papers having 466 citations to his work. He develops his authorship ObV option pricing. For many years he was a CEO and shareholder of Quant Technology, a company which practically implements ObV option pricing on the market. Now, he is mostly connected to blockchain technology. He is an advocate of social relations as the most important part of human life.

- **Did you have any prior experience that helped you succeed in this competition?**

  Andrzej has broad experience with programming. He joined kaggle 4 years ago and he was following and analysing many competitions from the past.

  Krzysztof's interests concentrate mainly on finance and algo trading. His practical experience in trading allowed us to understand some of the pitfalls of financial data. He is also an experienced Python programmer as well.

- **What made you decide to enter this competition?**

  About 4 years ago, Krzysztof was inspired by Andrzej to learn more about kaggle. At that time, we saw kaggle as a very interesting idea of an environment for data scientists and programmers. However, we started to be even much more engaged thanks to Two Sigma competition, first because winners got some honors to be one of the best from many other teams (now kaggle is widely recognized) as well as relatively prestigious awards for winners.

- **How much time did you spend on the competition?**

  We spent two weeks of very intense work for this competition. Krzysztof's role concentrated mainly on developing new solutions that came to his mind. As we worked remotely, we communicated through google hangout or skype calls to discuss in depth about new steps, approaches and their improvements on a daily basis. It was a very nice and funny Christmas time for both of us :).

- **If part of a team, how did you decide to team up?**

  We are cousins and we have been working together for many years. Before this specific competition, we were engaged in other projects. As our collaboration appeared to be very fruitful, we decided to continue to work together. We know each other well and what to expect from other team mates. We have common interests in programming but we have different experiences, which in the end generates synergy of our knowledge. Andrzej is engineer able to makes things done and Krzysztof focuses more on theoretical science.

- **If you competed as part of a team, who did what?**

  We both know programming, but Krzysztof is rather building the prototypes to check some dependency and Andrzej is an engineer able to implement optimized solutions. One of Andrzej's assets is the ease of independently learning new methodologies through internet or forums. He has a talent for picking up new ideas and skills as well as to implement them in practice. On the other hand, Krzysztof is very creative and generates new theories based on his proprietary knowledge rather than looking at others' solution. These two different approaches, seems to be common in the end and bring synergy which turns into a winning solution.

A3. Summary

To sum up, from our experience regarding algo trading, we came to the following conclusions regarding predicting financial prices:

- It is very probable to overfit financial model, so we would like to make some non-ML models or very simple ML models.

- We generally stick to the simple rule of thumb that the natural logarithm from number of data should be much less than the number of degrees of freedom. Here degrees of freedom can be understood as number of data features and parameters of ANN. That is why we did not artificially augment number of features and we kept the model simple:

ln(#data) >> #features

Our model in short summary:

- Model type:
  - ANN
- Most important features:
  - Volume
  - Close price
  - Open price
  - Returns from 1 and 10 days as raw data and residual to market
- The tools we used:
  - TensorFlow, Keras, Pandas
- Time needed to train our model:
  - up to 20 minutes

A4. Features Selection / Engineering

As above-mentioned, we did not artificially increase the number of features by any transformation and we removed news features from training. The reasons why we did not introduce news to generate signals, were the following:

- News parameters were too much subjective and depending on human judgement;
- Timing of news appearance can be misleading and can spoil news' importance;
- Numbers of news features were too big for a reliable simple model;
- Trails to introduce XGBoost to news features training were found to generate worse results compared to a whole ensemble model.

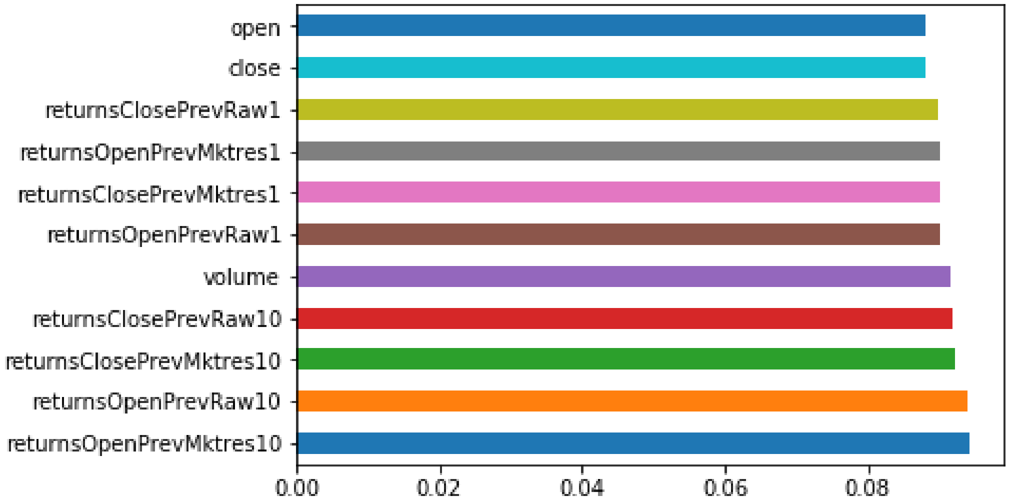

Fig.1 Features importance generated by ExtraTreesClassifier from module sklearn.

Fig. 1 presents a variable importance plot to be used together with a correlation heatmap (Fig. 2) in order to exclude variable with a very high cross correlation (dependency to each other). This can be used for choosing data features for a simplified model.

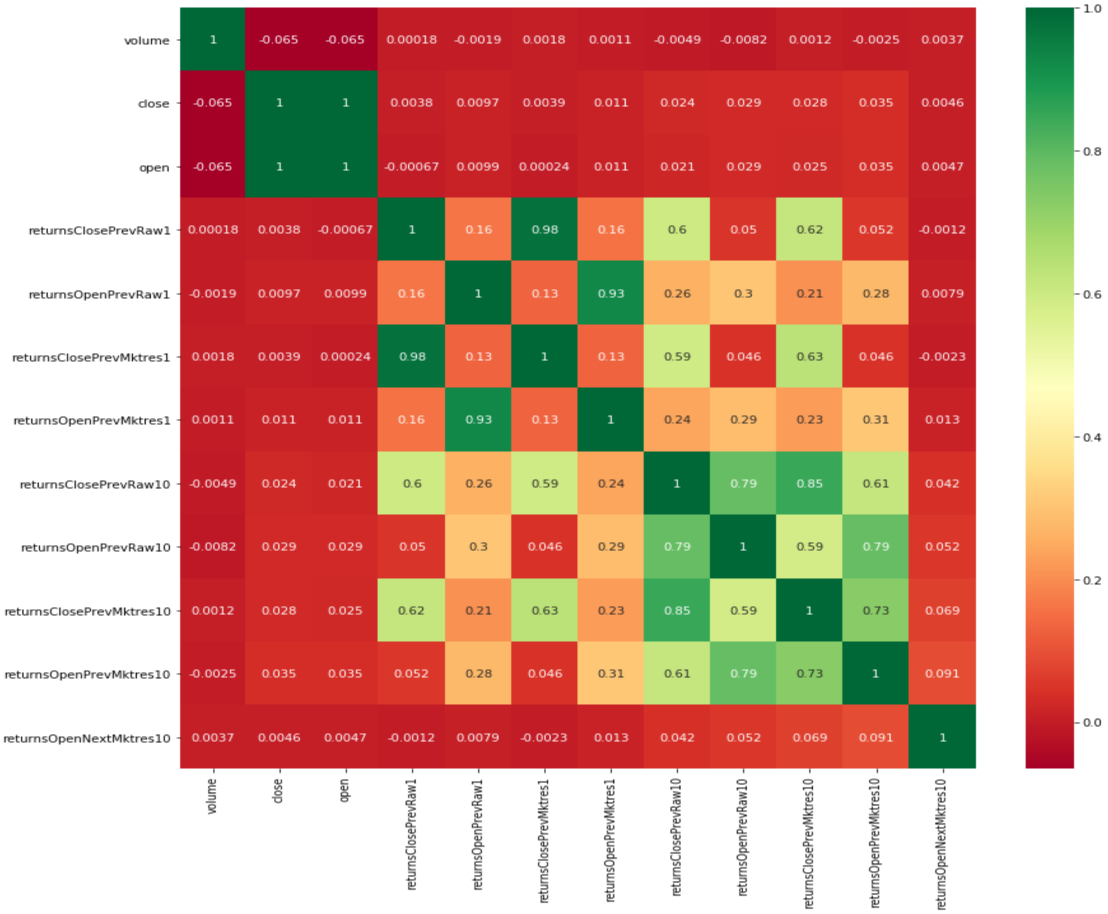

Fig. 2 Features cross-correlation heatmap.

Most important features include:

- **'returnsOpenPrevMktres10'** - positive correlation to target (9.1%) and with largest importance.
- **‘volume’** - low correlation to feature #1 and other features. Relatively high importance.
- **‘returnsOpenPrevRaw1’** - in the same manner selection was performed
- **‘close’** - the same reason as above-mentioned.

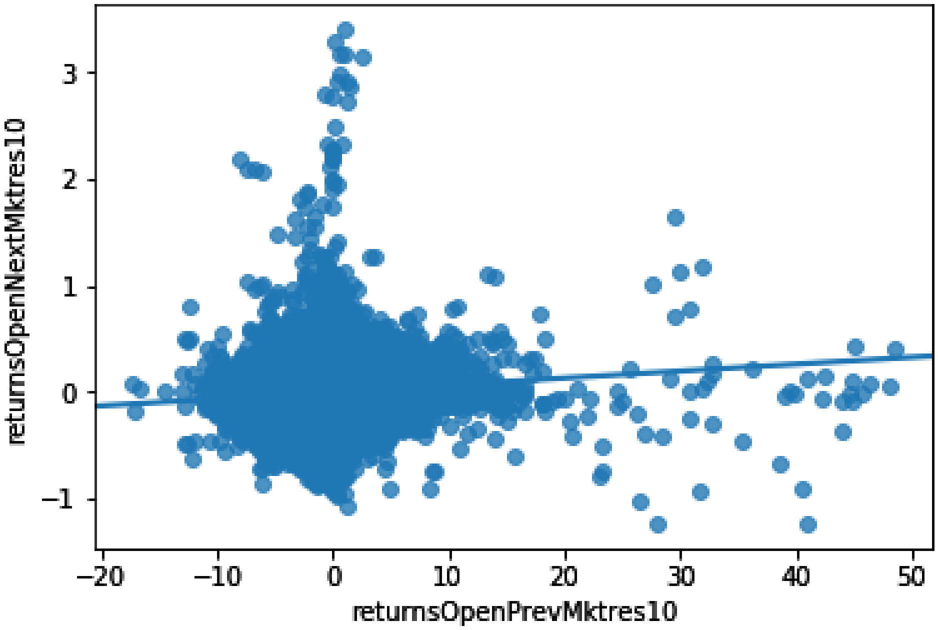

Fig 3. Relation and linear regression of the most important feature with the target.

A5. Training Method(s)

We use simple Artificial Neural Network with large dropout (0.4) and with max 32 neurons in hidden layers. We did not include L1/L2 regularization in our kernel. We tried to ensemble our model with XGBoost trained on news data, but finally we abandon this idea. In the end, we created a simple single ANN model. We did not use any hyperparameter tuning. The final model was trained with only 10 epochs with early stopping. The final ANN architecture follows as presented below in Fig. 4:

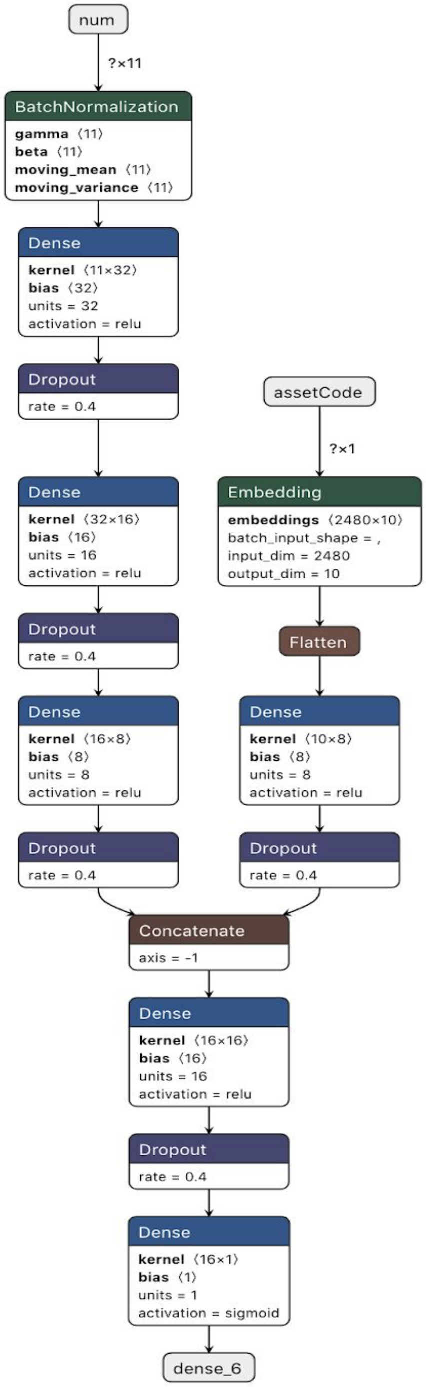

Fig. 4 Neural network architecture of a winning model.

A6. Interesting findings

What set you apart from others in the competition?

- Simplicity helps to avoid overfitting;
- Training was started from the beginning of 2015 year. As a financial system is structurally unstable, one should operate only on the most recent data;
- Custom loss function.

After many years of experience in trading, we observed that the past performance of algo has almost nothing to do with live evaluation and start trading, so we decided to keep our model simple and to test on data in a short time interval. The financial systems are rather structurally unstable, causing highly non-stationary time series. This often happens because in the financial systems are always involved people who may behave irrationally as their reaction to feedback in a neighbor space (other traders) and to the past. In that sense, psychological principles here are more valid than rational physics. Looking in the picture below, it can be seen that starting from 2018 price patterns seem to be sharper than ever before.

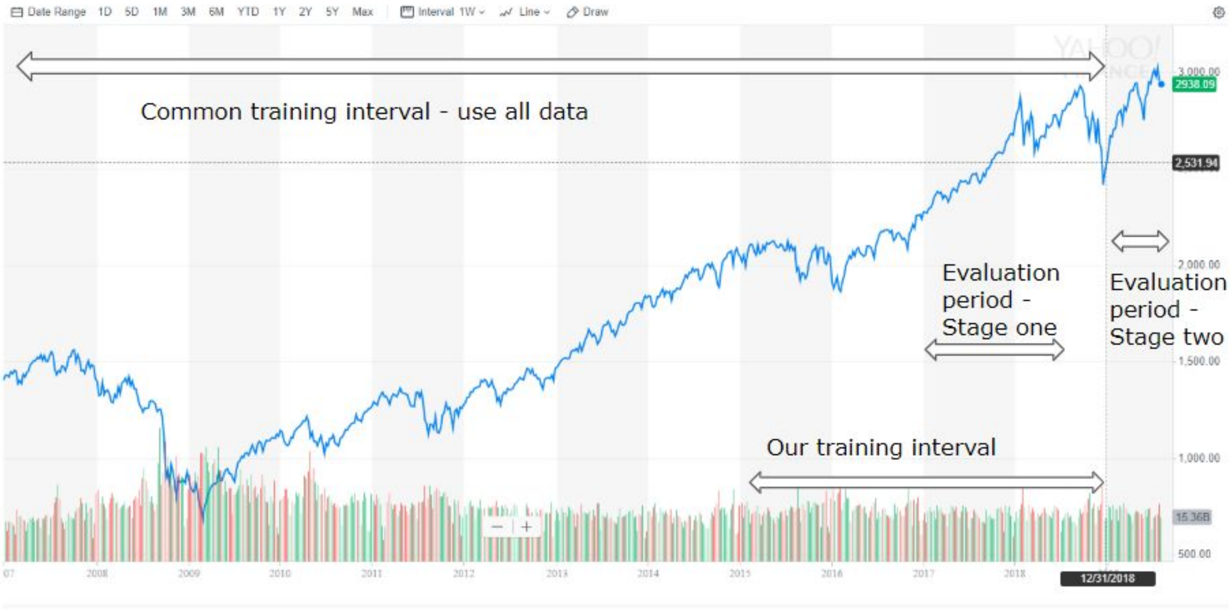

Fig. 5 Train and test period versus evaluation period based on S&P 500 index.

A7. Simple Features and Methods

Many customers are happy to trade off model performance for simplicity. Accordingly, below are some of our recommendations that are worth pursuing:

1. Please, forget about 3+ years past data. Train your model on the last 2-3 years data.
2. In custom objective function maximizes the simplified Sharpe ratio.
3. In order not to overfit, keep model simple. Please do not use rather ensemble models.
4. Reduce number of features to 4:
    - 'returnsOpenNextMktres10'
    - ‘volume’
    - ‘returnsOpenPrevRaw1’
    - ‘close’
5. Score for simplified model should be around: 0.6 - 0.8

A8. Model Execution Time

It takes just a few minutes to train a model, because of its simplicity. Simplified model will be also trained in the same or even a shorter time. Overall, the whole process should not take more than 20 minutes, depending on the hardware. Simplified version is supposed to take the same amount of time or a little bit shorter.

A9. References

1) Krzysztof Urbanowicz, “Informationism: from Philosophy to Quantitative Trading”, Amazon Digital Services LLC, August 29, 2015
2) http://www.quant-technology.com/
3) http://wonabru.org/