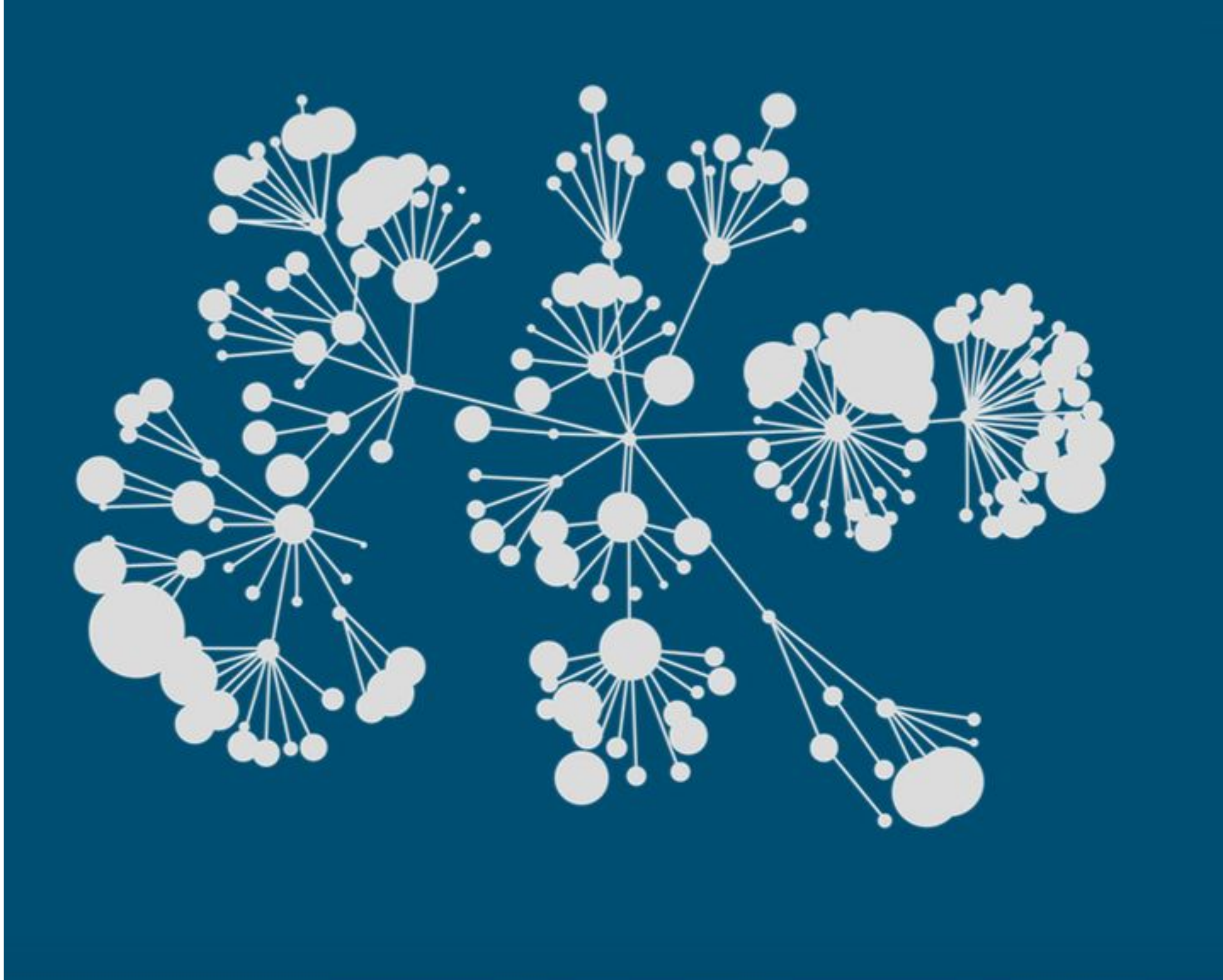
Kaggle

Two Sigma: Using News to Predict Stock Movements

Andrzej Urbanowicz
Krzysztof Urbanowicz



Agenda

1. Background
2. Summary
3. Feature selection & Engineering
4. Training methods
5. Important findings
6. Simple model

Background

Andrzej Urbanowicz

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

Background

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

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 that were used include:
  - TensorFlow, Keras, Pandas
- Time needed to train our model:
  - up to 20 minutes

Features Selection / Engineering

We generated several different approaches to cope with the problem and we found out what was the most important in this quest:

- It is very probable to overfit financial model, so we would like to make some non ML models or very simple ML models.
- We left only given features and did not generate new ones by any transformation.

Features Selection / Engineering

- We generally stick to the simple rule of thumb that the natural logarithm from number of data should be much less than number of degrees of freedom.

  ln(#data) >> #features

  Degrees of freedom can be understood as number of data features as well as of parameters of ANN. That is why, we did not artificially augment number of features by any transformation and we kept the model simple.

Features Selection / Engineering

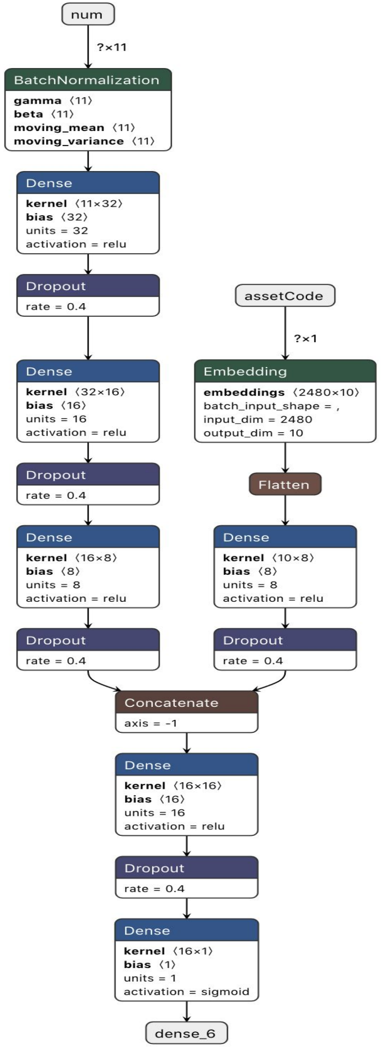
- Having in mind, that ML always tends to heavy fit to the train set, not to an unknown future as in this case, we allowed for this simple ANN a large dropout (0.4).
- For ANN construction, we set relatively small number of neurons with max 32 neurons in hidden layers.
- We did some simple preliminary cleaning of data (replacing with simple mean).
- We did not introduce L1/L2 regularization in ANN kernels.

Features Selection / Engineering

- We tried to introduce XGBoost to news features training, but we found out that it may worsen the results as a whole.

- We finally removed news from training, because:
  - it introduces too much noise;
  - it is 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.

ANN Architecture



num

?×11

BatchNormalization

gamma ⟨11⟩
beta ⟨11⟩
moving_mean ⟨11⟩
moving_variance ⟨11⟩

Dense

kernel ⟨11×32⟩
bias ⟨32⟩
units = 32
activation = relu

Dropout

rate = 0.4

Dense

kernel ⟨32×16⟩
bias ⟨16⟩
units = 16
activation = relu

Dropout

rate = 0.4

Dense

kernel ⟨16×8⟩
bias ⟨8⟩
units = 8
activation = relu

Dropout

rate = 0.4

assetCode

?×1

Embedding

embeddings ⟨2480×10⟩
batch_input_shape = ,
input_dim = 2480
output_dim = 10

Flatten

Dense

kernel ⟨10×8⟩
bias ⟨8⟩
units = 8
activation = relu

Dropout

rate = 0.4

Concatenate

axis = -1

Dense

kernel ⟨16×16⟩
bias ⟨16⟩
units = 16
activation = relu

Dropout

rate = 0.4

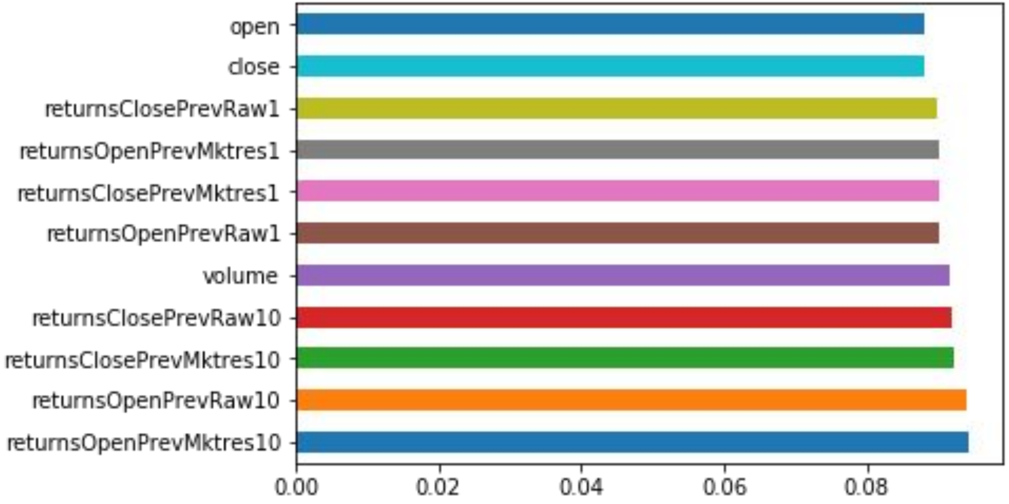
Dense

kernel ⟨16×1⟩
bias ⟨1⟩
units = 1
activation = sigmoid

dense_6

Features Selection / Engineering

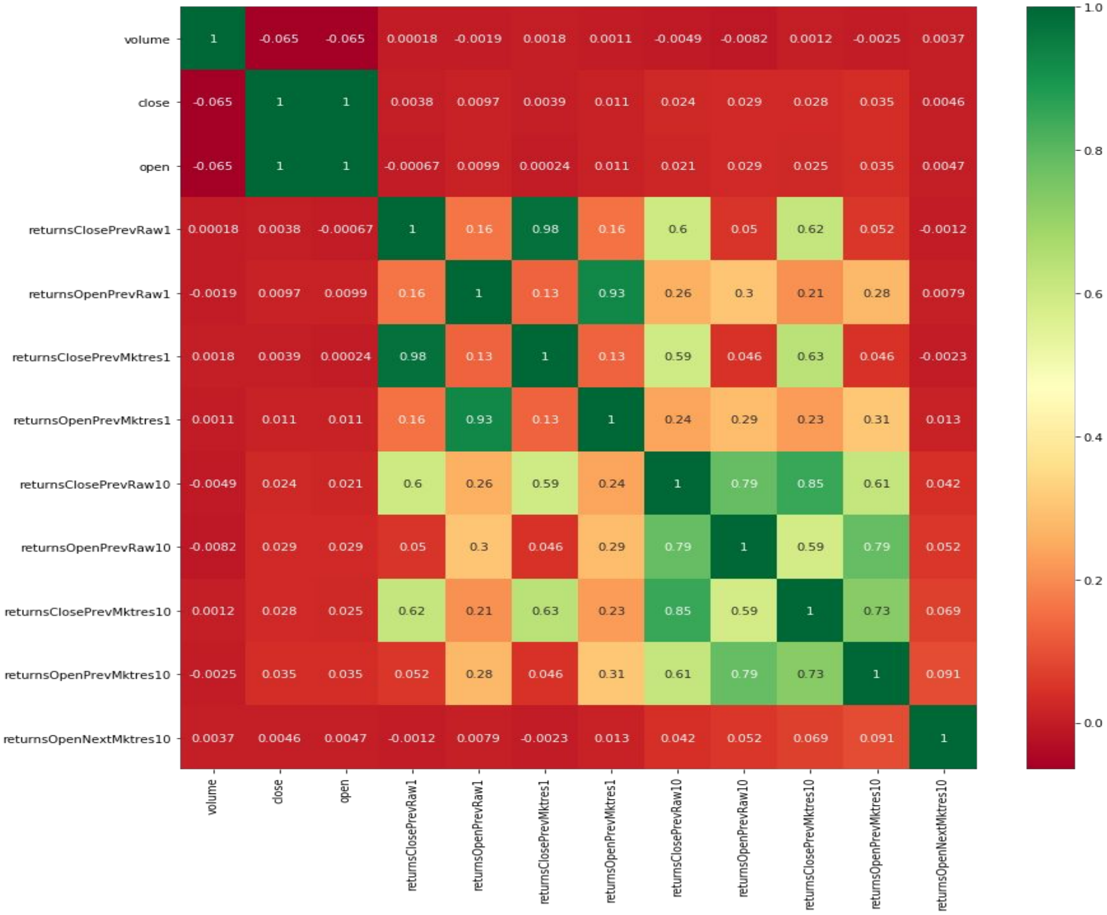
Variable Importance Plot



Variable importance plot to be used together with a correlation heatmap (see next slide) 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.

Features Selection / Engineering

Variable Cross-Correlation Heatmap



	volume	close	open	returnsClosePrevRaw1	returnsOpenPrevRaw1	returnsClosePrevMktres1	returnsOpenPrevMktres1	returnsClosePrevRaw10	returnsOpenPrevRaw10	returnsClosePrevMktres10	returnsOpenPrevMktres10	returnsOpenNextMktres10
volume	1	-0.065	-0.065	0.00018	-0.0019	0.0018	0.0011	-0.0049	-0.0082	0.0012	-0.0025	0.0037
close	-0.065	1	1	0.0038	0.0097	0.0039	0.011	0.024	0.029	0.028	0.035	0.0046
open	-0.065	1	1	-0.00067	0.0099	0.00024	0.011	0.021	0.029	0.025	0.035	0.0047
returnsClosePrevRaw1	0.00018	0.0038	-0.00067	1	0.16	0.98	0.16	0.6	0.05	0.62	0.052	-0.0012
returnsOpenPrevRaw1	-0.0019	0.0097	0.0099	0.16	1	0.13	0.93	0.26	0.3	0.21	0.28	0.0079
returnsClosePrevMktres1	0.0018	0.0039	0.00024	0.98	0.13	1	0.13	0.59	0.046	0.63	0.046	-0.0023
returnsOpenPrevMktres1	0.0011	0.011	0.011	0.16	0.93	0.13	1	0.24	0.29	0.23	0.31	0.013
returnsClosePrevRaw10	-0.0049	0.024	0.021	0.6	0.26	0.59	0.24	1	0.79	0.85	0.61	0.042
returnsOpenPrevRaw10	-0.0082	0.029	0.029	0.05	0.3	0.046	0.29	0.79	1	0.59	0.79	0.052
returnsClosePrevMktres10	0.0012	0.028	0.025	0.62	0.21	0.63	0.23	0.85	0.59	1	0.73	0.069
returnsOpenPrevMktres10	-0.0025	0.035	0.035	0.052	0.28	0.046	0.31	0.61	0.79	0.73	1	0.091
returnsOpenNextMktres10	0.0037	0.0046	0.0047	-0.0012	0.0079	-0.0023	0.013	0.042	0.052	0.069	0.091	1

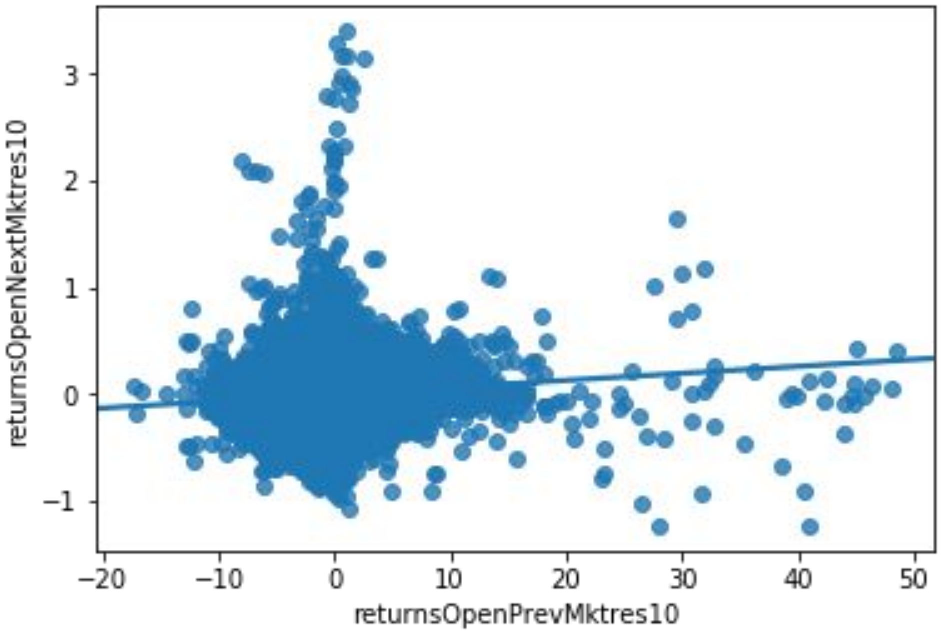
Features Selection / Engineering

Most important features:

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

Features Selection / Engineering

Relation and linear regression of the most important feature with the target



Training Methods

- Training methods:
  - used only data from 2015-01-01
  - 75% data used for training
  - 10 epochs
  - EarlyStopping callback
  - not used hyperparameter tuning
  - Adam optimizer with custom loss function
- Did you ensemble?
  - No. Try to keep model as simple as possible in order not to overfit.
  - Ensembling current model with news model generated always the worst results.

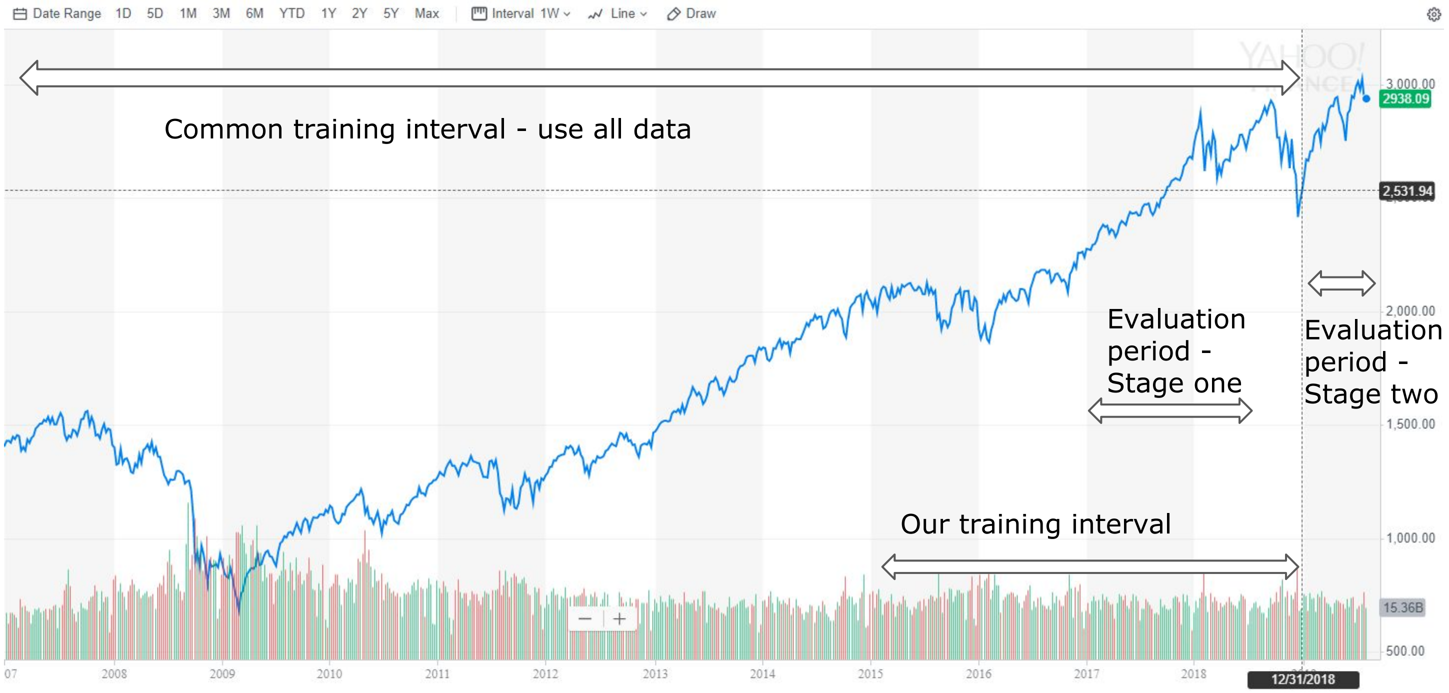
Important and Interesting Findings

- What set you apart from others in the competition?
    - Simplicity;
    - Training was started from beginning 2015 year. Financial system is structurally unstable, so 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 test on data in short time interval.

Important and Interesting Findings

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



Common training interval - use all data

Evaluation period - Stage one

Evaluation period - Stage two

Our training interval

Source: Yahoo Finance

Looking in the picture above, it can be seen that starting from 2018 price patterns seem to be sharper than ever before.

Simple Model

- A subset of features that could contribute to reach 90-95% of your final performance:
  - Please, forget about 3+ years past data. Train your model on the last 2-3 years data.
  - In custom objective function maximizes the Sharpe ratio.
  - In order not to overfit, keep model simple. Please, rather do not use ensemble of models.
  - Reduce number of features to 4 (see slide 13th).
  - Score for simplified model should be around: 0.6 - 0.8.

Thank you very much!

kaggle™