What we learned from Kaggle Two Sigma News Competition

Ernie Chan, Ph.D. and Roger Hunter, Ph.D.

QTS Capital Management, LLC.

The Competition

- Kaggle hosts many data science competitions
  - Usual input is big data with many features.
  - Usual tool is machine learning (but not required).
- Two Sigma Investments is a quantitative hedge fund with AUM > $42B.
  - Sponsored Kaggle news competition starting Sept, 2018, ending July, 2019.
  - Price, volume, and residual returns data for about 2,000 US stocks starting 2007.
  - Thomson-Reuters news sentiment data starting 2007.
  - Evaluation criterion: Sharpe ratio of a user-constructed market (beta)-neutral portfolio*.

Our Objectives

- Does **news sentiment** generate alpha?
  - Find out using normally expensive, high quality data.
- Does **machine learning** work out-of-sample?
- Does successful ML == successful trading strategy?
- How best to **collaborate** in a financial data science project?
- Educational: example **lifecycle** of trading strategies development using data science and ML.

Constraints

- All research must be done in cloud-based Kaggle kernel using Jupyter Notebook.
  - Only 4 CPU's, limited memory and slow.
  - Kernel killed after a few idle hours.
  - Cannot download data for efficient analysis.
  - Cannot upload any supplementary data to kernel (E.g. ETF prices).
  - Poor debugging environment (it is Jupyter Notebook!)
  - Lack of "securities master database" for linking stocks data.

Features

- *Unadjusted* open, close, volume, 1- and 10-day raw and *residual returns*.
  - Jonathan Larkin[1] designed PCA to show that

    residual returns = raw returns - β* market returns

    = CAPM residual returns

    [1] www.kaggle.com/marketneutral/eda-what-does-mktres-mean
- News sentiment, relevance, novelty, subjects, audiences, headline, etc.
  - Numerical, categorical, textual.

Target and Evaluation Criterion

- Target(t, s): Open-to-open 10-day residual return from day t+1 to t+11 for stock s (given features available up to 23:59:59 UTC on day t.)
- Prediction(t, s): Predicted sign(Target(t,s))
- Pos(t, s): Prediction(t, s)*Capital_Weight(t, s)
- Evaluation: Winner has highest

$$score = \frac{mean(\sum_s \mathrm{Target(t,s)} * \mathrm{Pos(t,s)})}{std(\sum_s \mathrm{Target(t,s)} * \mathrm{Pos(t,s)})}$$

=Sharpe Ratio of **zero-beta** portfolio of stocks **hedged with market index**.

Data Issues and Cleansing

- Lack of “securities master database” – need to create our own unique id (**uid**).
  - Otherwise impossible to merge price and news data!
- Need to create our own split/dividend adjustment price series for “fractional differentiation” [2].

  [2] Lopez de Prado, “Advances in Financial Machine Learning”
- Bad price data prior to 2009.
- How do we know if there are errors in news data?

Creating uid

- assetName = company name
  - assetName of a company already set to its most recent by data vendor.
- assetCode = ticker symbol
- Many assetCode → One assetName
- One assetName → Many assetCodes
- T-Mobile → (PCS.N, TMUS.N, TMUS.O)
  - Ticker changes over time.
  - Red ticker is most recent assetCode, used as our **uid**!
- Alphabet → (GOOG.O, GOOGL.O)
  - 2 classes of stocks co-exist.
  - Need to differentiate them due to different price (but not news) data!

Creating uid

- If two assetCodes for same assetName co-existed contemporaneously
  - Use both as **uid**s.
- If two assetCodes for same assetName *didn't* co-exist contemporaneously
  - Just a ticker change.
  - (We checked price and time gap to confirm this.)
  - Use most recent assetCode as **uid**.

Bad Price Data

- Kagglers' consensus: Many errors before 2009.
- Kagglers[3] checked all returns, and changes of prices and volumes over threshold.

  [3] www.kaggle.com/danielson/cleaning-up-market-data-errors-and-stock-splits
- They replace bad open, close, volume with correct.
  - Correct numbers from outside sources.
- They interpolate residual returns.
- We clip target residual returns to [-1,1]

News Data Errors

- Time series plots of statistics of numerical news features *show no structural breaks*.
- No obvious way to check categorical features.

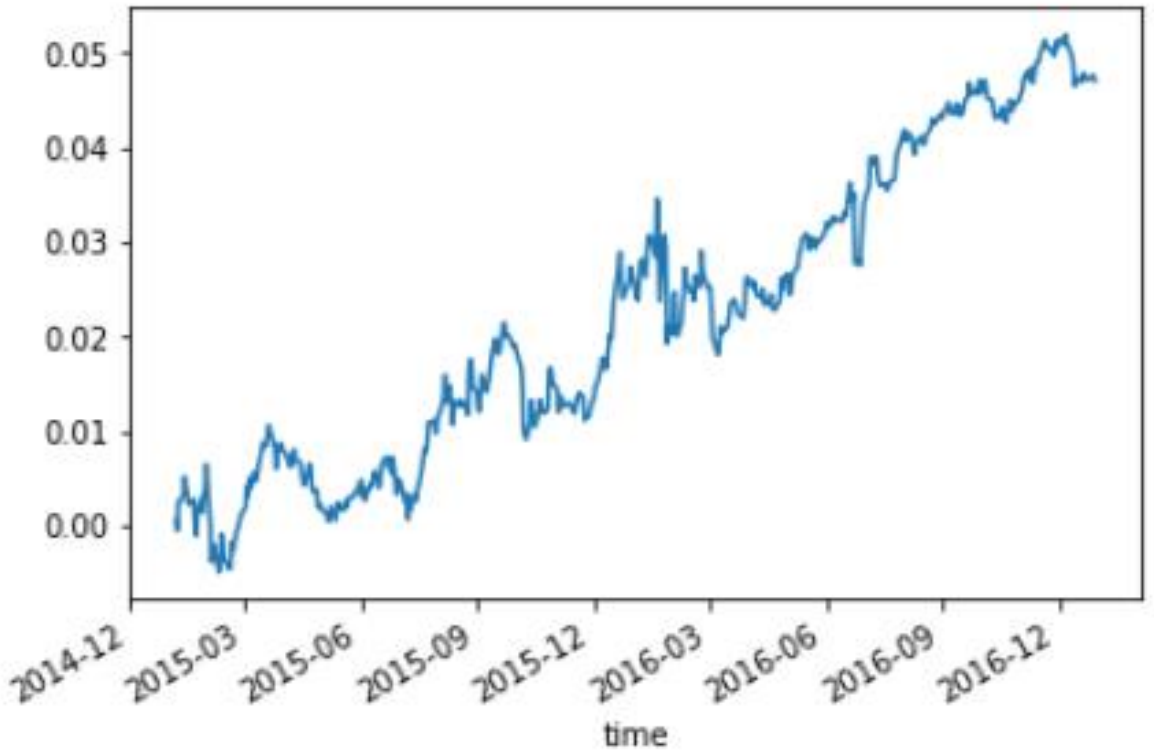
News Features

- 2 important numerical features:
  - Sentiment ([-1, 1])
  - Relevance ([-1, 1])
- We combine these features and take 5-day moving average of product: movavg(s*r)
- Prediction(t, s) = $\begin{cases} +1 \text{ if movavg}(s*r) > 0 \\ -1 \text{ if movavg}(s*r) < 0 \end{cases}$

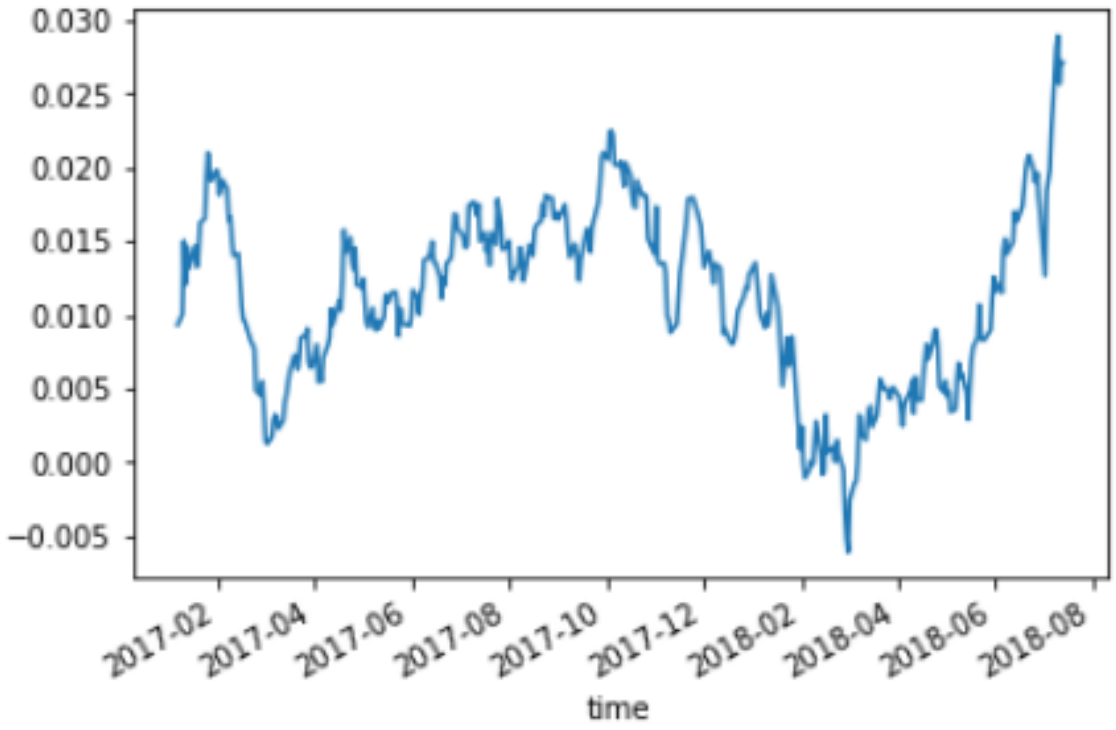
Naïve News Strategy

- Buy and hold for 10 days if Prediction(t, s)=+1
- Short and hold for 10 days if Prediction(t, s)=-1
- Hedge any beta exposure with market index.
- Equal capital allocation.
- Result on validation set:
  - CAGR=2.3% (“alpha”)
  - Sharpe Ratio=1
- Result on test set:
  - CAGR=1.8% (“alpha”)
  - Sharpe Ratio=0.75

News Strategy: validation set



News Strategy: test set

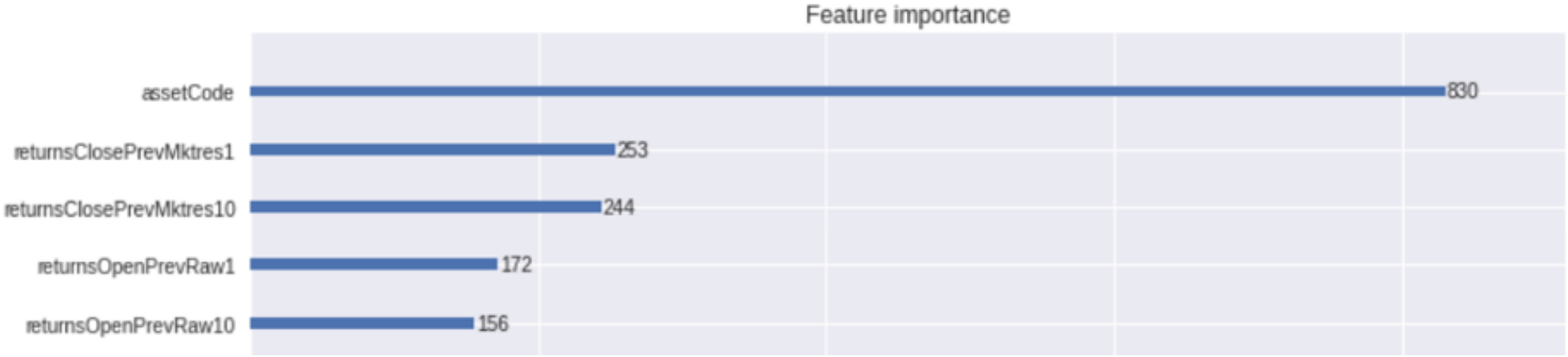


Categorical News Features

- Single value: E.g. headlineTag=‘BUZZ’
- Set of values: E.g. audiences={'O', 'OIL', ‘Z’}
- E.g. headlineTag has 163 unique values, audiences has 191 .
- Ordinal feature or one-hot encoding?
- Many stocks have multiple rows per day.
- Combine daily features with one-hot and OR.
- Use LightGBM for features selection.

Features Selection

- Problem with LightGBM feature importance
  - Uses training data, not validation data
  - Hence assetCode and assetName are picked [4]!



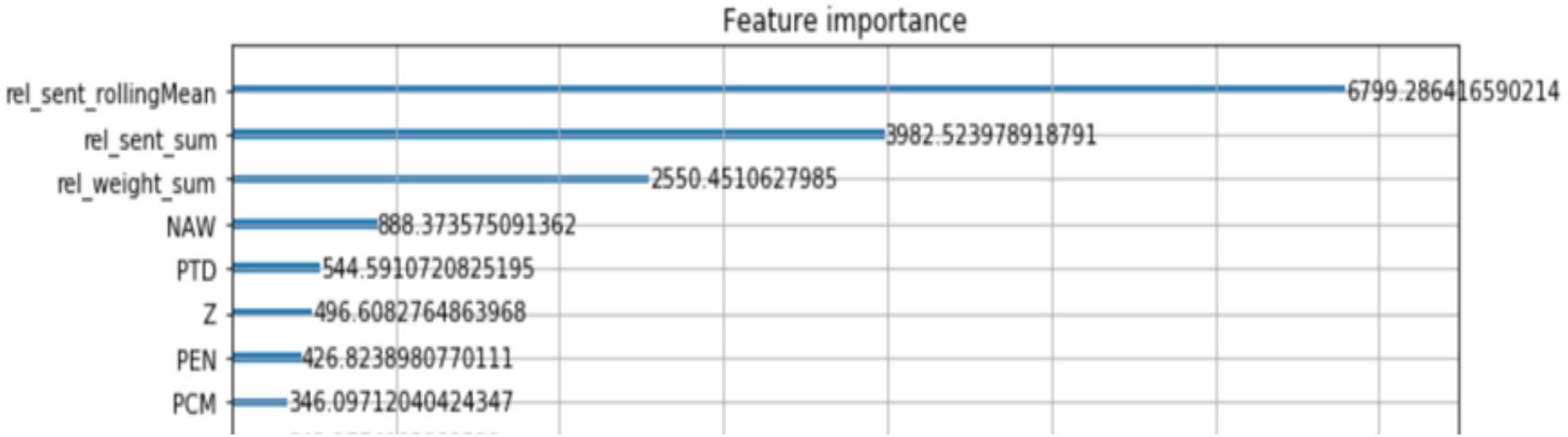
[4] Larkin, www.kaggle.com/marketneutral/https://www.kaggle.com/marketneutral/the-fallacy-of-encoding-assetcode

- Solutions: MDA (CV or OOS) [5] or use non-constant features.

[5] Chan and Hunter, www.kaggle.com/chanep/assetcode-with-mda-using-random-data

Audiences

- Use only 50 most common categorical values



- headlineTag, etc. similarly unimportant.

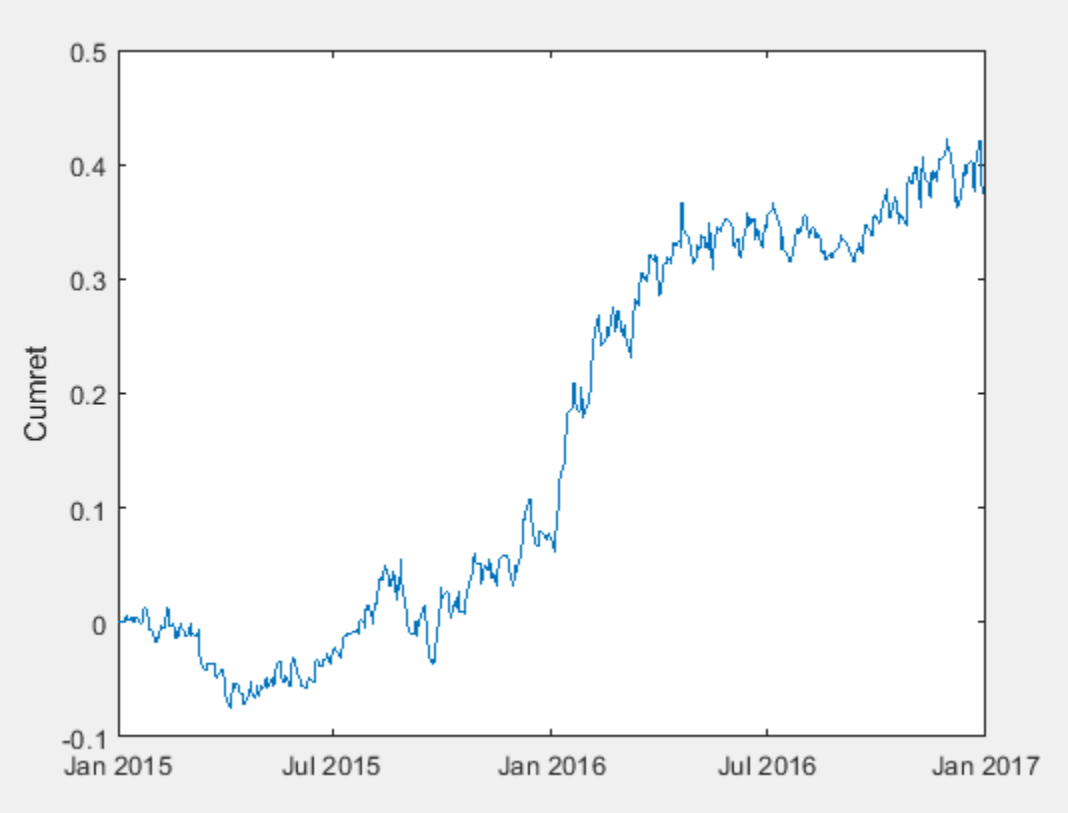
Price Features

- We have also created simple features based on prices and volumes only. For e.g.
  - Past 10-day residual returns.
  - Lagged past 10-day residual returns.
  - Fractionally differentiated price series.
  - Change in volume.
- Use logistic with L1/L2 regularizations to predict signs of future returns.
- Capital allocation: “risk parity”
  - Inversely proportional to past volatility of returns.

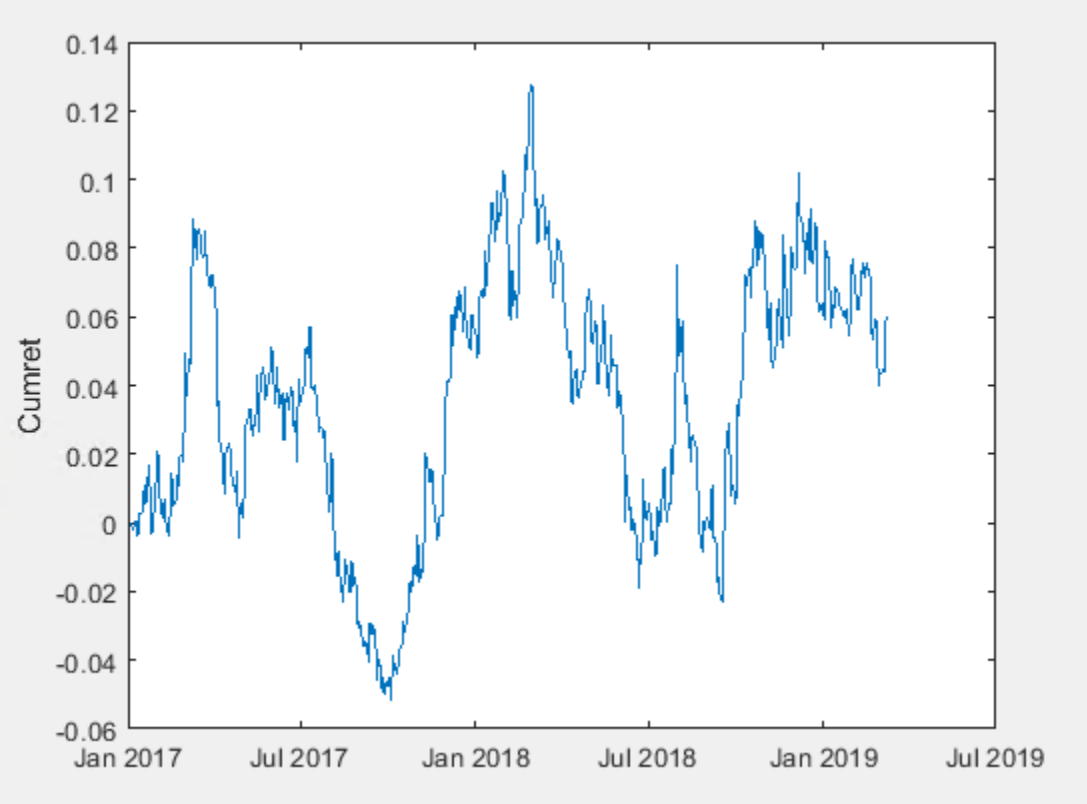
Price Strategy

- Result on validation set:
  - CAGR= 17.2% (“alpha”)
  - Sharpe Ratio= 1.2
- Result on test set:
  - CAGR= 2.7% (“alpha”)
  - Sharpe Ratio= 0.28

Price Strategy: validation set



Price Strategy: test set



Conclusion

- For both news and price strategies, alpha and Sharpe ratio significantly lower in test set than validation set.
- News strategy does not require training and hence little scope for overfitting.
  - Large “variance” likely due to alpha decay of news sentiment.
  - Beckers, 2018 (JPM) meta-study of news sentiment research found average information ratio of news sentiment strategies to be less than 0.5 from 2008-2017! (Performance roughly $^1/_2$ of 1998-2017.)
- Price strategy’s Sharpe ratio deteriorated more in test set.
  - Likely due to overfitting, despite simple, regularized logistic regression model.
  - We can’t rule out regime change either.
  - Simple technical features do not work.
  - Insights into specific market inefficiencies still required!

Download my talks at

www.epchan.com