Summary

January 26, 2021

Table of Contents

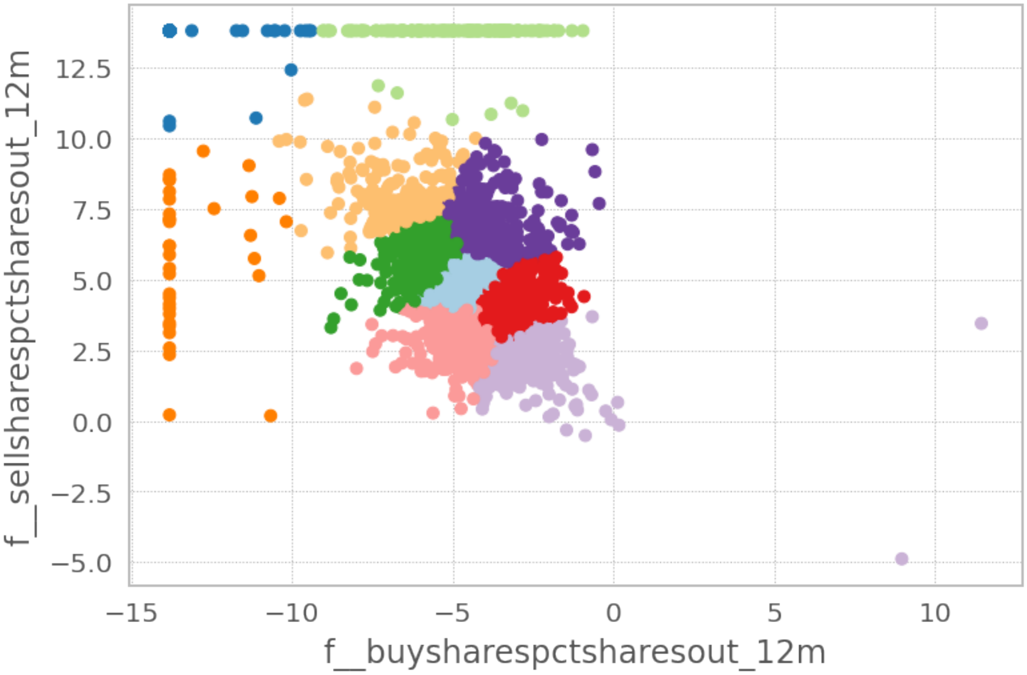
1 EDA

- Data is reported on Fridays, with a single report per Company (if available) per given day
- Data is more or less evenly distributed for test (Jan 2016 – Sep 2019) and train (Aug 2004 – Dec 2015)
- Reporting periods for a given Company may span from 7 days (consistent weekly reporting) to any multiple of 7 (gaps from a week to a year+)
- A gap doesn't guarantee there was no reporting in the period (a 9m gap –> report of an insider trading within 6m –> no current or previous report for a shorter period)
- Number of reports per Company id spans from 1 to 195 in test and to 595 in train
- 9695 stocks in train and 4966 in test
- 24% of Companies in test don't have history in train
- 112 Companies in test are single datapoint, with 59 out of them not having history in train
- There are some interesting artefacts in the data like longer reporting periods seem to positively correlate to better outcomes and other, that need to be explained and incorporated into modeling

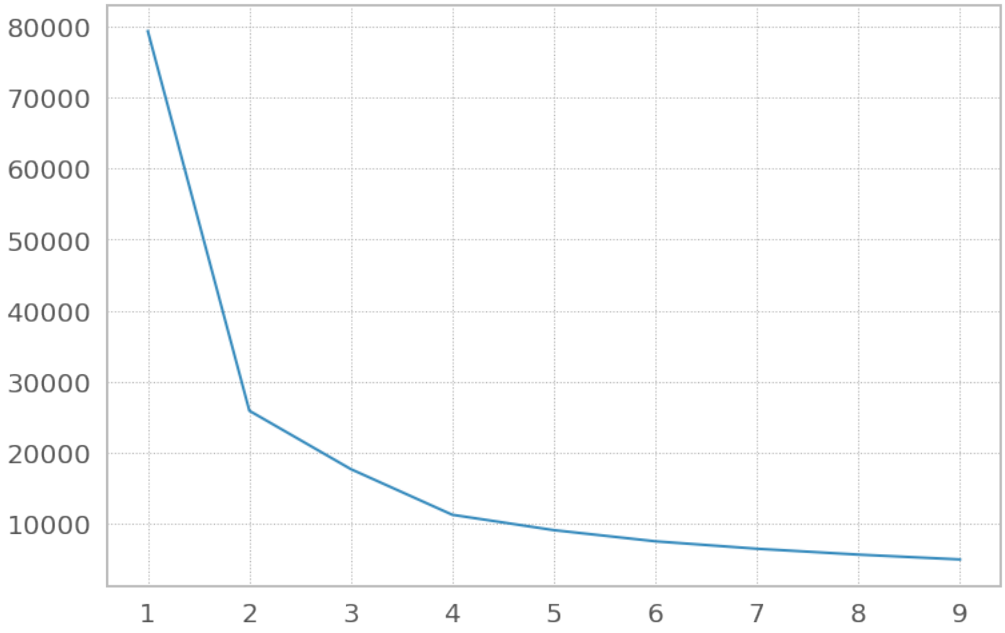
2 Feature engineering

- In addition to original features the following features were suggested:
  - **Counts** of historical reportings available on a day of prediction
  - **Target mean encoding**. For a certain datapoint we embed a feature representing historical 1m, 3m, 6m, 12m performance. Due to (i) the way target is constructed, i.e. it's 12m forward looking, and (ii) a 3 year ahead performance prediction was asked, target encoding was shifted 3 years back, i.e. we are embedding 3y-1m, 3y-3m, 3y-6m, 3y-1m target means, which is obviously suboptimal.

- **Linear combinations of original features**:
  - 1m to 1m, 3m to 3m, 6m to 6m, 12m to 12m differences of sell vs buy features
  - 1m differences to 3m, 6m, 12m sell/buy metrics
- **Clustering**. Stocks seem to belong to highly heterogeneous groups, with sell/buy counts ranging from low single digits to multiples of 10^{15}. Thus clustering was suggested to put similar stocks together. Clustering was done on every possible feature pair, after features normalization, which ensures even stock distribution among different clusters:



- 4 to 10 clusters seem a fair choice for optimal num of clusters for this particular pair (10 was chosen for all clusters):



3 Cross validation

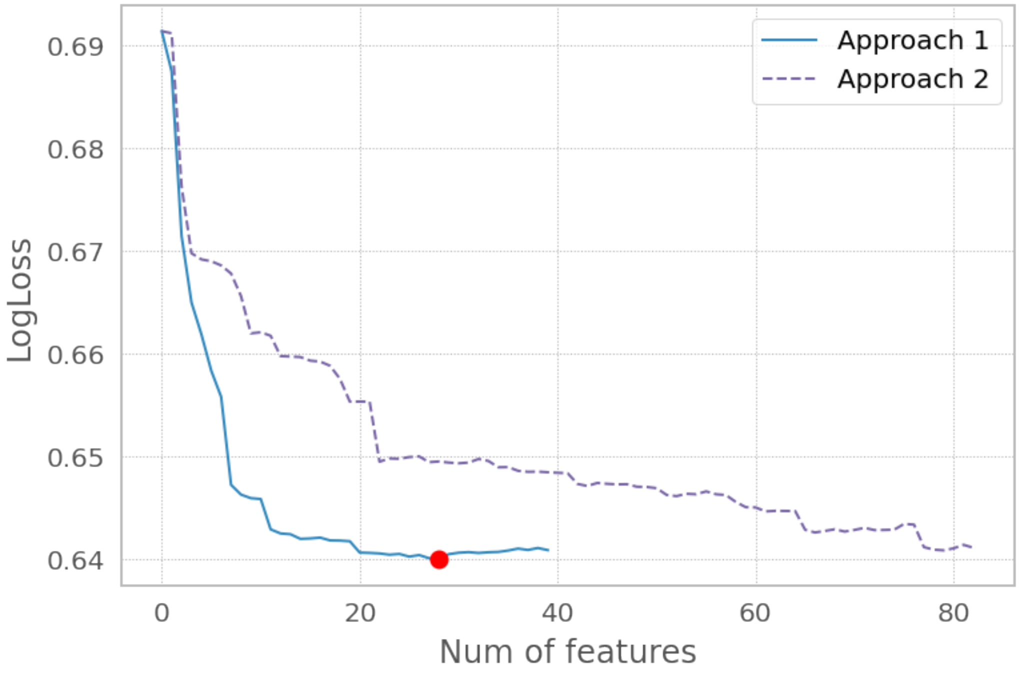
- For the best case exercise data with at least 3 year history was chosen, which accounts for 50% of the whole data.
- To test for model generalization ability the data was further split into Train (<2010) and Test (2011,2012, 2013) folds (which may easily be generalized to a 5 fold time expanding CV)
- Target mean encoding was done on the whole dataset, all the rest feature engineering was put into a **pipeline**:
  - Feature transformations are **learnt** (fit method) and **applied** (transform method) on a train fold.
  - **Only transformations** are applied on a test

4 Model tuning

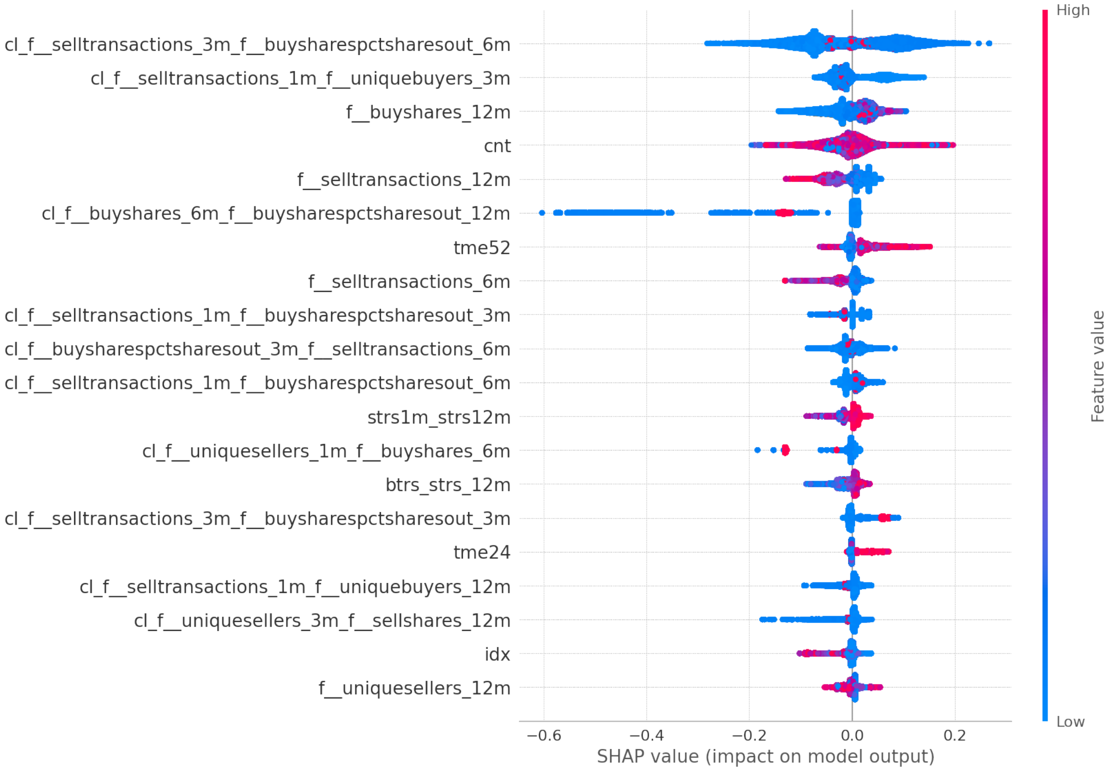
- LightGBM classifier was tuned for best hyperparams on the ability to generalize to test fold.

5 Feature selection

- With 500+ features many of them may exhibit collinearity, multicollinearity, or other types of non-linear interdependence, which may hinder model's ability to learn.
- Different methods were tried to get the most parsimonious model with a satisfactory performance



- Choosing best feature subset on SHAP values seem to provide the best model (from 579 features to 21, logloss from 0.6836 to 0.6838, 1 fold, out-of-sample)



6 What else could be tried

1. Projected over-/under-performance (current target) exhibit autocorrelation so adding features showing recent performance makes sense
2. Clustering on recent 1m, 3m, 6m, 12m sell/buy history
3. Different normalization strategies prior to clustering
4. Adding features comparing sell/buy features to those of competitors' on prediction date or recent history
5. Adding fast/slow MA crossovers on original sell/buy features