Multi-step feature selection for forecasting commodity trading price movements

Adam Harasimowicz
ForecastThis Inc., Cambridge, United Kingdom
adam@forecastthis.com

Abstract: Feature selection has a great impact on model performance. Most of the attribute picking methods belong to one of the three families – wrappers, filters or embedded approaches. The paper shows the usability of different feature selection methods in predicting a direction of commodity trading price movements. Also, a method for adapting these selectors to work with high-dimensional data is presented as well as a universal framework for generating and selecting features for forecasting different kinds of commodities.

Keywords: feature selection, machine learning, market forecasting, time series

1. Introduction

In machine learning domain, the feature selection (FS) is an important part of model building process. Choosing the right attributes allows reducing problem complexity and required computations to find proper solution by decreasing dimensionality of a hypothesis search space [39]. However, selecting wrong or not enough features may lead to obtaining worse models – in terms of methods used for evaluation as well as future performance.

The majority of the feature selection methods tries to remove redundant or irrelevant information. It is especially important for high-dimensional input data where both kinds of information are commonly observed [39]. Additionally, choosing relevant information is not a trivial task because even attributes useless by themselves can bring usable information when they are incorporated with other features [13]. Moreover, not all of the general purpose choosing attributes approaches are suitable for time series where consecutive samples are highly bound to each other [3, 34].

There are three main strategies for feature selection which cover the majority of used methods. The first of them, wrapping, focuses on finding a proper feature set with considering the interactions between chosen learning algorithm and the selected subset of attributes. In this approach, the modeling algorithm is used as a black box by the feature selection process and the model performance is used to measure a relevance of the selector’s choice [24]. In practice, original train set is usually split into two subsets during cross-validation or holdout process. These datasets are used as a new train set and validation set to perform feature selection. After that, the chosen features are extracted from the original sets, and are used for final learning and evaluation [13]. Depending on implementation, different methods for looking through potential feature space subsets can be applied – like greedy search [24], simulated annealing [25, 28] or genetic algorithm [17, 41].

The next strategy, filtering, can be performed in two different ways – supervised and unsu-
Multi-step feature selection for forecasting commodity trading price movements

In both cases, to find the proper subset of features, we analyze a general characteristic of information to assess the feature importance. Also, for supervised approaches, relations between target and features are checked. Then, based on the importance, feature ranking is prepared indicating which attributes have the strongest discriminative power. During assessment different metrics can be used – like cross entropy \[33\], Fisher score \[12\], t-test \[16\], information gain \[10\], constraint score \[40\], area under a ROC curve \[7\] or Laplacian score \[15\]. In the filtering approach, we usually omit learning models during selection in opposite to strategies based on wrapping. Moreover, many of the filtering methods do not consider a joint discriminative power of the attributes what often means that wrapping strategy provides a better solution. However, it requires greater computational resources than filtering. Therefore, sometimes hybrid strategies are preferred \[19, 32\].

The last strategy is embedding in which feature selection is linked to fitting predictor as in wrapping. However, this connection is much stronger because choosing attributes is included into model construction \[26, 36\]. Nonetheless, the feature selection is not the main goal, but rather a side effect of ensuring that models will remain robust after training. Embedded methods are usually specific to particular learning algorithms \[13\]. Well-established examples of such methods are logistic regression with regularization \[29\], Random Forest \[4, 5, 9\] or Gradient Boosted Trees \[30, 37\].

Working with market data is a good example of dealing with high-dimensional time series. For price- or volume-based features we can generate a vast amount of descriptive and derivative features including common indicators from technical analysis widely used in trading. In such way that we can easily obtain tens of thousands attributes. Besides, there are other sources of information which often are unstructured and can be also an origin of the new features \[38\].

Forecasting changes in the market is a difficult task. In addition to high dimensionality, the data are intensive, noisy, non-stationary and with hidden relationships, a high degree of uncertainty as well as often have unstructured nature \[1, 18\]. The three dominant algorithms for forecasting are a family of autoregressive methods \[1\], Support Vector Machine \[18, 20, 21\] and Artificial Neural Network \[1, 2, 6, 20, 22, 31\]. On the other hand, the feature selection is mostly the manual process which is based on domain expert knowledge and prior researches \[1, 6, 20, 21, 22, 31\]. Nevertheless, the feature selection methods described earlier, are sometimes also used. Such as filtering strategy containing modified Relief algorithm and correlation-based analysis \[2\] or TDIDF-based input document selection \[38\].

In this paper, we describe different feature selection methods – including general approaches and dedicated for the time series. We also measure and compare their influence on the performance of the predictors for market modeling as well as we assess how similar feature subsets to each other they produce. Besides, we present the universal framework for forecasting trading prices of diverse commodities and the direction of their price movements. The proposed solution includes multi-step hybrid feature selection.

2. Feature selection methods

For the experiments, seven different kinds of algorithms have been chosen. Those are representatives of the all described above selection strategies. Also, there are general methods as well as specialized which should work well with time series.
Correlation-based Feature Selector

CFS (Correlation-based Feature Selector) employs search algorithm along with a heuristic evaluation function to measure goodness of the feature subset. Methods for searching may vary depending on implementation. Hall mentions three algorithms – backward elimination, forward selection and best first [14]. For our experiments, we used the last one. To evaluate subset of selected features CFS uses the following function (Equation 1):

\[ M_S = \frac{k \bar{r}_{tf}}{\sqrt{k + k(k - 1) \bar{r}_{ff}}} \]

where \( M_S \) is a heuristic merit of the subset, \( k \) is a number of attributes, \( \bar{r}_{tf} \) and \( \bar{r}_{ff} \) are mean target-feature correlation and average feature-feature inter-correlation, respectively. The numerator of Equation 1 expresses relevance of the selected subset and the denominator indicates how much redundancy there is among the attributes.

In summary, CFS is the computationally inexpensive method which is able to quickly identify irrelevant, noisy and redundant features. However, it is not designed to find features whose relevance strongly depends on other attributes [14].

Consistency-based Filter

Consistency-based Filter tries to retain the discriminative power of the original data, but with a smaller set of attributes. Algorithm defines the consistency rate which is used to measure a quality of the selected subset. As an inconsistency, which is consistency’s opposition, are interpreted samples with the same feature values, but with different labels. The consistency rate is a ratio of consistent patterns to all of them. The way how patterns are counted is described in [8].

In the same paper, five different methods for searching subsets are presented – exhaustive, complete, heuristic, probabilistic and hybrid. To perform our experiments, we use heuristic strategy due to its speed and ability to work with large attribute sets.

Consistency-based measures heavily rely on training data and prefer hypotheses definable over feature subsets as small as possible [8].

Random Forest

Random Forest (RF), opposite to the two previously presented algorithms, is an example of embedded feature selection. In RF, variable importance can be measured in different ways. Commonly used metrics, to assess the quality of a split and variable usefulness, are Mean Decrease Accuracy and Gini importance [27].

In the experiments, we use a selection method which utilizes importance spectrum. Based on variable importance, obtained from RF, general feature ranking is prepared as well as separated ordered lists for datasets with permuted class labels. Given these rankings, final feature selection occurs [11].

Relief

Relief algorithm avoids the heuristic search for feature selection. Instead, it relies on statistical method. It also works in linear time in the size of the data matrix. The algorithm is designed to work with binary classification problems and requires scaled data. The method starts with vector \( W \) of size \( p \) filled by zeros where each element is a weight of the particular
attribute. Then, during every iteration, it chooses random sample \( X \) and using \( p \)-dimensional Euclid distance looks for the most similar samples in the rest of the data – one for every class. Because the method works with binary classification problems, we denote these samples as \( X_{hit} \), \( X_{miss} \) for the sample which belongs to the same class like \( X \) and to the opposite class, respectively. Then, using Equation 2, we update the weight for every feature \( i \).

\[
W_i = W_i - (X_i - X_{hit,i})^2 + (X_i - X_{miss,i})^2
\]  

(2)

It means that if \( X_{hit} \) is close to \( X \), the weight is not decreased significantly. Also, the weight mostly increases if the found sample for the opposite class is not similar to \( X \). Finally, based on vector \( W \) and given threshold, Relief selects the most accurate features according to its policy [23].

**Logistic Regression**

Logistic Regression (given by Equation 3), as the Random Forest, can be considered as an embedded method for feature selection. After training the model, we can treat absolute values of coefficients as a measure of the attributes importance.

\[
h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}
\]  

(3)

We can also employ regularization to obtain sparse results. \( L1 \) method (Equation 4) has some advantages over \( L2 \). It requires fewer samples in a train set with the same number of features to solve a problem. This is especially visible if the data contain a large number of attributes because \( L1 \)’s sample complexity is logarithmic in the number of features when for \( L2 \) it is linear. Also, \( L1 \) leads to more sparse results than \( L2 \) [29].

\[
h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} + \lambda \| \theta \|_1
\]  

(4)

**Progressive Correlation Method**

Progressive Correlation Method is an example of wrapping strategy [34]. On contrary to the previous methods, this algorithm was designed to work with time series and it favors, during selection, features with the least time lags when there are more features with comparable relative correlation coefficients. It also prefers a smaller group of chosen attributes if several groups have similar modeling residuals. The time lag can be viewed as a time distance which describes from how far in the past information is used.

The algorithm starts with selecting the first feature which is the attribute with the smallest time lag among variables whose correlation level with target exceeded some threshold. Then, noisy features are removed. Afterwards, the model is trained and new attributes are added progressively to the set of chosen features until root mean squared error stops decreasing or all features have been checked. Features are selected based on the model residual series and relative correlation coefficients.

The whole procedure is repeated a few times with different values of the threshold. Then, from subsets of the features with small modeling residuals, we select the one which has the shorter time lag and fewer attributes – as mentioned earlier [34].
Pearson’s Correlation-based Selector

We also decided to use a filter for feature selection which employs Pearson’s correlation as the similarity measure. Such approach is simple and computationally efficient. Besides, this metric is commonly used for comparing time series. During the first step, we check the absolute correlation between target and features and keep attributes which exceed some threshold found based on quantiles. Then, for all features which survived the initial selection, we make a relevance ranking. Next, features are chosen iteratively in the order scheduled by ranking. Before the feature will be accepted, its correlation with the already selected attributes is checked to prevent choosing redundant features.

Besides, we prepared another variant of the described method with modified similarity measure. In such a way, that among features with similar correlation with the target, attributes with a smaller number of missing values are preferred (Eq. 5). Introducing this change reflects a statistical significance of the calculated feature importance which is proportional to square root of the time series length [35].

\[
importance(x, y) = \sqrt{n}(\rho(x, y))^2
\]

where \( x \) is a feature, \( y \) is a target, \( n \) – series length and \( \rho \) is a Pearson’s correlation function.

3. Data processing and feature selection

To address issues related to high-dimensional data, we designed our system to iteratively add new features and perform feature selection. Also, to speed up computation, we parallelized the most time-consuming parts as well as used aggressive data shrinking.

The process of generating dataset for the model starts with retrieving data from remote and local sources (phase A in a figure 1). Then, all data are unified and merged by timestamps. Our base data mostly contain OHLCV information (open, high, low, close prices and value of traded volume) for over 150 different financial instruments. For them, we calculate different indicators from technical analysis – approx. 65 for each instrument (phase B). It means that after this step, we have more than nine thousand features in the dataset. Then, we remove OHL data as they will not be useful anymore and are highly redundant (phase C). The next step is transforming data into logarithmic space (D).

Data processing, described so far, is common for all datasets – regardless of what instrument is the target of forecasting. The next steps are target-specific. We start with shrinking data by removing old information when target instrument has not existed yet – for this step we keep time margin defined by longest time range required by derivative features which will be computed later (phase E). Also, we remove features which do not contain the most recent information – as they will not be useful when models are applied to real forecasting. On average, the shrinking allows decreasing data size twelve times.

In the next step, we continue data shrinking but by performing the first feature selection (F). Choosing proper attributes is done based on absolute values of Pearson’s correlation between prices for target instrument and the rest of the data. A relatively low threshold is used during this phase to screen out only these variables which are very unlikely to provide useful information for the future model.

On the contrary, next phase leads to increase the size of the dataset. We add features which contain price changes for different time horizons as well as descriptive attributes like minimum, maximum, mean value and standard deviation (phase G). Twelve kinds of ranges
are used – starting from daily changes and ending with seven months price differences. It causes a six-fold growth of the data meaning that the dataset contains over twenty thousand attributes.

Then, the second step of feature picking takes place. Used attribute selection technique vary and is chosen from methods described in section 2 (phase H). Because different methods can use diverse metrics and scales to express feature importance, we use a value of a 75th percentile as threshold – if such kind of parameter is required by the selector.

Thereafter, six different low-pass filters are used to add new features to the dataset (phase I). This step causes more than two-fold increase in the number of the attributes.

Finally, last variable selection is performed (phase J). In this step, the more rigorous threshold is used – the 95th percentile value. Both threshold values, for current and previous attribute picking steps, were chosen as the result of preliminary experiments and the trade-off between dimensionality and data expressiveness. Similar logic was behind choosing the places where feature selections occur. To say it more precisely, these values and places were chosen in such way, that in any point of the data transformation process, the total number of features do not exceed $2.5 \times 10^4$ and the number of selected attributes into the final feature set is in a range between 100 and 300. These boundaries were established because of the relation between required resources and searching space size as well as the quality of forecasts of modeling algorithm, respectively. After this step, data are ready for modeling.

Table 1 presents the number of features in the dataset after every described phase. These values are valid for the building dataset for natural gas for one-week forecasts with Pearson’s Correlation-based Selector. Numbers of features for other kinds of targets and selectors are similar.

### 4. Selecting features in high-dimensional space

During processing pipeline implementation (described in section 3) and performing preliminary experiments, we decided to drop the Consistency-based Filter for its poor ability to discriminate features in our forecasting problem. This caused a not big enough space
Table 1. Number of features after every processing step

<table>
<thead>
<tr>
<th>Phase</th>
<th>FS</th>
<th>No FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>772</td>
<td>772</td>
</tr>
<tr>
<td>B</td>
<td>9175</td>
<td>9175</td>
</tr>
<tr>
<td>C</td>
<td>8252</td>
<td>8252</td>
</tr>
<tr>
<td>D</td>
<td>8252</td>
<td>8252</td>
</tr>
<tr>
<td>E</td>
<td>5579</td>
<td>5579</td>
</tr>
<tr>
<td>F</td>
<td>3531</td>
<td>5579</td>
</tr>
<tr>
<td>G</td>
<td>22439</td>
<td>35454</td>
</tr>
<tr>
<td>H</td>
<td>5610</td>
<td>35454</td>
</tr>
<tr>
<td>I</td>
<td>13561</td>
<td>85703</td>
</tr>
<tr>
<td>J (choosing relevant features)</td>
<td>678</td>
<td>85703</td>
</tr>
<tr>
<td>J (removing redundant features)</td>
<td>260</td>
<td>85703</td>
</tr>
</tbody>
</table>

size reduction, implying that the problem quickly started to be unmanageable due to limited resources.

Also, many of the selected methods were not able to work efficiently with high-dimensional data, e.g. due to their computational complexity higher than linear. To resolve it, we proposed a split-select-merge framework based on divide and conquer paradigm, where data are randomly divided into smaller groups and for the each group feature selection is applied. After it, all selected features are merged into one set. The whole process is repeated a few times and the number of iterations depends on obtained reduction of the feature set size. Used sizes of groups and number of iterations are presented in table 2. They were chosen using the same principles and boundaries like for the threshold.

Table 2. Group sizes for feature selection methods which use split-select-merge framework

<table>
<thead>
<tr>
<th>Method</th>
<th>Phase</th>
<th>Group size</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>H</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>Random Forest</td>
<td>H</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>550</td>
<td>2</td>
</tr>
<tr>
<td>Relief</td>
<td>H</td>
<td>1000</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>H</td>
<td>500</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td>500</td>
<td>2</td>
</tr>
</tbody>
</table>

5. Missing values and data quality

There are plenty reasons for the introduction of missing values into financial data. Some of those values are the result of merging data about different instruments which were created at different times. Another of them, by holidays in countries from which data are used. Also, some unusual situations, like closing stock exchange market due to political or economic reasons, can be their source. Because some machine learning algorithms are not able to work with missing values, we filled empty places by the last known value for considered time series at particular points in time.

Modeling movements of commodity trading prices require restrictive policies to make sure that model evaluation is valid. It means that data, as well as models, should be prepared and
trained in a streaming fashion. Also, any information which was not available at the given
time cannot be used. To follow this rule, it is important to know precisely from which time
zones data are gathered. Breaking time order during data processing or modeling, may result
in overoptimistic model evaluation which will not be consistent with real forecasting power.

Besides, the feature selection can introduce some bias to the modeling process. For this
reason, looking for useful attributes should be performed on separated portion of the data from
the information used for evaluation, thus preventing overfitting.

To identify any inadvertent leakage of future information within tens of thousands po-
tential derivative features, we introduce simple but useful test. First, we choose arbitrary
time point \( T \) and two identical datasets \( S_1 \) and \( S_2 \) with OHCLV information about financial
instruments. Next, we split set \( S_1 \) at time point \( T \) and keep all samples with \( t < T \). Then,
we perform data processing, feature generation and selection described in section 2 for each
set. After it, we split \( S_2 \) also at time point \( T \) in the same manner like \( S_1 \) and compare both
remaining subsets. If they are identical, it means that there are not any future information
leakage among features.

6. Results

During the experiments described earlier, data processing and feature selection process
was run multiple times with the incorporation of different feature selection techniques. To
measure overlap between results, we used coverage metric described by Equation 6.

\[
coverage(A, B) = \frac{|A \cap B|}{|B|}
\]  

(6)

Table 3 presents mean coverage between feature selection methods across two kinds of
targets – natural gas and aluminium, and five different time horizons – one and two days,
week, half a month and one month. The results indicate that in the most cases feature selection
methods choose substantially different subsets of attributes. The only one exception is related
with Pearson’s Correlation-based Selectors as they are similar methods to each other.

<table>
<thead>
<tr>
<th>A</th>
<th>CFS</th>
<th>LogReg</th>
<th>Relief</th>
<th>RF</th>
<th>PCM</th>
<th>PCS</th>
<th>PCS LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>1</td>
<td>0.02</td>
<td>0.01</td>
<td>0.02</td>
<td>0.05</td>
<td>0.03</td>
<td>0.06</td>
</tr>
<tr>
<td>LogReg</td>
<td>0.05</td>
<td>1</td>
<td>0.03</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>Relief</td>
<td>0.07</td>
<td>0.05</td>
<td>1</td>
<td>0.08</td>
<td>0.08</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>RF</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
<td>1</td>
<td>0.06</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>PCM</td>
<td>0.1</td>
<td>0.02</td>
<td>0.02</td>
<td>0.04</td>
<td>1</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>PCS</td>
<td>0.04</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>1</td>
<td>0.61</td>
</tr>
<tr>
<td>PCS LS</td>
<td>0.05</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.42</td>
<td>1</td>
</tr>
</tbody>
</table>

Groups of selection methods – \( A \) and \( B \) – correspond to sets in
Eq. 6. Table contains results for following methods: Correlation-based
Feature Selector (CFS), Logistic Regression (LogReg), Relief, Ran-
dom Forest (RF), Progressive Correlation Method (PCM), Pearson’s
Correlation-based Selector (PCS), Pearson’s Correlation-based Selector
with favoring features with less missing values (PCS LS).
Then, we used selected feature subsets by different methods to train gradient-based classifiers in streaming fashion. Models had to forecast the direction of price movements. For all targets, predictions were made for a seven-year period. Table 4 presents rounded values of accuracy across all targets. On average, the best results were provided by Progressive Correlation Method. Also, Pearson’s Correlation-based Selector chose strong features. However, depends on the target, different selectors give better accuracies. Besides, because chosen subsets are quite different, we suppose that merging attribute subsets or ensembling models can significantly improve the predictive power of classifiers.

To obtain more robust and solid comparison of the usefulness of the feature selection methods for predicting commodity trading price movements, we decided to conduct additional experiments. The phase F from the figure 1 (preliminary feature selection) was replaced by random feature sampling. In such a way that 2000 features were chosen during phase F. Also, the whole modified data processing pipeline was executed 15 times. The obtained results are presented in table 5. For every execution, the table shows mean accuracy across both instruments and all forecasting time horizons.

In the second round of experiments, Pearson’s Correlation-based Selector and Logistic Regression achieved the best results. Surprisingly, Progressive Correlation Method – the strongest approach in the first round – provided almost the worst features for learning models. Also, the most of the results obtained during the second round of experiments are lower than previously. Being the consequence of random sampling which can easily drop important features.

Table 4. Accuracy

<table>
<thead>
<tr>
<th>Natural Gas</th>
<th>CFS</th>
<th>LogReg</th>
<th>Relief</th>
<th>RF</th>
<th>PCM</th>
<th>PCS</th>
<th>PCS LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>0.48</td>
<td>0.52</td>
<td>0.49</td>
<td>0.49</td>
<td>0.51</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>2 days</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.49</td>
<td>0.51</td>
<td>0.51</td>
<td>0.52</td>
</tr>
<tr>
<td>1 week</td>
<td>0.53</td>
<td>0.53</td>
<td>0.49</td>
<td>0.52</td>
<td>0.58</td>
<td>0.55</td>
<td>0.56</td>
</tr>
<tr>
<td>2 weeks</td>
<td>0.55</td>
<td>0.58</td>
<td>0.54</td>
<td>0.58</td>
<td>0.59</td>
<td>0.62</td>
<td>0.52</td>
</tr>
<tr>
<td>1 month</td>
<td>0.58</td>
<td>0.58</td>
<td>0.59</td>
<td>0.59</td>
<td>0.62</td>
<td>0.56</td>
<td>0.53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aluminium</th>
<th>CFS</th>
<th>LogReg</th>
<th>Relief</th>
<th>RF</th>
<th>PCM</th>
<th>PCS</th>
<th>PCS LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>0.5</td>
<td>0.51</td>
<td>0.48</td>
<td>0.51</td>
<td>0.49</td>
<td>0.51</td>
<td>0.51</td>
</tr>
<tr>
<td>2 days</td>
<td>0.5</td>
<td>0.51</td>
<td>0.51</td>
<td>0.53</td>
<td>0.51</td>
<td>0.51</td>
<td>0.51</td>
</tr>
<tr>
<td>1 week</td>
<td>0.5</td>
<td>0.55</td>
<td>0.51</td>
<td>0.51</td>
<td>0.56</td>
<td>0.53</td>
<td>0.5</td>
</tr>
<tr>
<td>2 weeks</td>
<td>0.54</td>
<td>0.5</td>
<td>0.51</td>
<td>0.52</td>
<td>0.54</td>
<td>0.53</td>
<td>0.5</td>
</tr>
<tr>
<td>1 month</td>
<td>0.59</td>
<td>0.52</td>
<td>0.52</td>
<td>0.5</td>
<td>0.57</td>
<td>0.59</td>
<td>0.55</td>
</tr>
<tr>
<td>Mean</td>
<td>0.53</td>
<td>0.53</td>
<td>0.51</td>
<td>0.52</td>
<td>0.55</td>
<td>0.54</td>
<td>0.52</td>
</tr>
</tbody>
</table>

7. Conclusions

In the paper, we presented that general feature selection methods can be adapted to work with high-dimensional time series through employing divide and conquer approach. Nevertheless, the results can indicate that some of the feature selection methods are better than others when used with time series. Also, described feature calculating framework is universal and can be used for any kind of commodities. Thus, we would like to propose two directions for the future researches. The first of them is extending the subset of commodities with other instruments. The second has been already mentioned earlier in the paper and is related to combining the subsets of features received from different selection methods.
Table 5. Accuracy for experiments with random sampling

<table>
<thead>
<tr>
<th>Execution No.</th>
<th>CFS</th>
<th>LogReg</th>
<th>Relief</th>
<th>RF</th>
<th>PCM</th>
<th>PCS</th>
<th>PCS LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>0.526</td>
<td>0.516</td>
<td>0.510</td>
<td>0.515</td>
<td>0.531</td>
<td>0.529</td>
</tr>
<tr>
<td>2</td>
<td>0.529</td>
<td>0.530</td>
<td>0.523</td>
<td>0.525</td>
<td>0.524</td>
<td>0.525</td>
<td>0.511</td>
</tr>
<tr>
<td>3</td>
<td>0.530</td>
<td>0.529</td>
<td>0.515</td>
<td>0.523</td>
<td>0.511</td>
<td>0.533</td>
<td>0.525</td>
</tr>
<tr>
<td>4</td>
<td>0.520</td>
<td>0.528</td>
<td>0.512</td>
<td>0.524</td>
<td>0.521</td>
<td>0.530</td>
<td>0.524</td>
</tr>
<tr>
<td>5</td>
<td>0.523</td>
<td>0.530</td>
<td>0.520</td>
<td>0.528</td>
<td>0.513</td>
<td>0.533</td>
<td>0.521</td>
</tr>
<tr>
<td>6</td>
<td>0.533</td>
<td>0.528</td>
<td>0.516</td>
<td>0.524</td>
<td>0.506</td>
<td>0.534</td>
<td>0.523</td>
</tr>
<tr>
<td>7</td>
<td>0.530</td>
<td>0.531</td>
<td>0.516</td>
<td>0.527</td>
<td>0.517</td>
<td>0.520</td>
<td>0.528</td>
</tr>
<tr>
<td>8</td>
<td>0.527</td>
<td>0.529</td>
<td>0.508</td>
<td>0.524</td>
<td>0.510</td>
<td>0.528</td>
<td>0.523</td>
</tr>
<tr>
<td>9</td>
<td>0.520</td>
<td>0.526</td>
<td>0.510</td>
<td>0.520</td>
<td>0.524</td>
<td>0.532</td>
<td>0.531</td>
</tr>
<tr>
<td>10</td>
<td>0.531</td>
<td>0.526</td>
<td>0.505</td>
<td>0.528</td>
<td>0.520</td>
<td>0.531</td>
<td>0.518</td>
</tr>
<tr>
<td>11</td>
<td>0.520</td>
<td>0.535</td>
<td>0.519</td>
<td>0.524</td>
<td>0.521</td>
<td>0.520</td>
<td>0.525</td>
</tr>
<tr>
<td>12</td>
<td>0.528</td>
<td>0.529</td>
<td>0.518</td>
<td>0.519</td>
<td>0.513</td>
<td>0.537</td>
<td>0.523</td>
</tr>
<tr>
<td>13</td>
<td>0.520</td>
<td>0.526</td>
<td>0.518</td>
<td>0.526</td>
<td>0.526</td>
<td>0.536</td>
<td>0.538</td>
</tr>
<tr>
<td>14</td>
<td>0.520</td>
<td>0.532</td>
<td>0.520</td>
<td>0.525</td>
<td>0.519</td>
<td>0.539</td>
<td>0.523</td>
</tr>
<tr>
<td>15</td>
<td>0.523</td>
<td>0.530</td>
<td>0.515</td>
<td>0.522</td>
<td>0.517</td>
<td>0.527</td>
<td>0.519</td>
</tr>
<tr>
<td>Mean</td>
<td>0.525</td>
<td>0.529</td>
<td>0.515</td>
<td>0.523</td>
<td>0.517</td>
<td>0.531</td>
<td>0.524</td>
</tr>
</tbody>
</table>

Lastly, we want to mention that having a model, which is able to achieve better accuracy than random choice, does not guarantee having profitable predictor. Due to transactional fees which can vanish possible profits.

References


