Machine Learning – A Machine's Perspective on Positioning

Mark Keenan
Head of Research and Strategy at Engelhart Commodity Trading Partners; and Editorial Advisory Board Member, *Global Commodities Applied Research Digest*

Book Overview and Summary

Advanced Positioning, Flow and Sentiment Analysis in Commodity Markets is a new book focusing on positioning dynamics in commodities. The book covers substantial new material, but also updates and builds significantly on some of the work in the previous book, *Positioning Analysis in Commodity Markets – Bridging Fundamental and Technical Analysis*, by Mark Keenan. New material includes analytics based on the analysis of flow, the decomposition of trading flows, trading activity in the Chinese commodity markets, the inclusion of newsflow into Positioning Analysis and how machine learning can provide insight into trading relationships.

Wiley Trading Advantage
ADVANCED POSITIONING, FLOW, AND SENTIMENT ANALYSIS IN COMMODITY MARKETS
BRIDGING FUNDAMENTAL AND TECHNICAL ANALYSIS
MARK J S KEENAN
WILEY

Behavioral patterns driven by positioning and flow dynamics can change and evolve as different types of market participants enter and leave the market and as new price drivers emerge, which can lead to the formation of new patterns and relationships. The book provides new and alternative ways of thinking about commodity markets with new tools and analytics to help understand them better, track how these relationships evolve to improve trading performance, and risk management.

The ideas, insights, and concepts behind the signals, indicators, and models in the book have been developed over the last 20 years throughout a variety of different market conditions and regime changes. In many cases, their construction is unique, but in all cases, the approach is robust, intuitive, and accessible to commodity market participants and risk managers on a variety of levels and in different areas of the market.

This digest article is based on the final chapter of the book on machine learning. It introduces decision trees and random forests as ways of potentially uncovering relationships between changes in positioning and changes in commodity prices.

The objective is to use a machine to identify which aspects of positioning are the most useful in helping to understand commodity markets from a machine's perspective.

It shows that machine learning is particularly useful in the analysis of positioning data, with "feature importance" a powerful way of identifying new patterns and new relationships in positioning.

The results provide alternative insights that can help improve how other positioning signals, indicators, and models are interpreted and used.

Introduction to Machine Learning (ML)

> Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.
>
> Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.
>
> Machine learning algorithms are used in the applications of email filtering, detection of network intruders, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task.
>
> Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning.
>
> Data mining is a field of study within machine learning and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.
>
> Source: https://en.wikipedia.org/wiki/Machine_learning

The main objective of Machine learning is to uncover predictive relationships within datasets. It is broadly divided into two areas: supervised learning and unsupervised learning. In supervised learning, an algorithm is first calibrated (or trained) on a dataset to identify relationships between a group of input variables (X) and an output variable (Y). In unsupervised learning, the algorithm seeks to identify patterns within the input variables.

In this article, decision trees (classification and regressions trees) and the random forest algorithm (classification and regression random forests) are introduced as ways of uncovering relationships between changes in positioning and changes in commodity prices - specifically to learn which aspects of positioning are the most useful in helping to better understand commodity markets, from a machine learning perspective.[1]

Tree-based learning algorithms – including decision trees and random forests – are amongst the most-used learning methods and can map non-linear relationships easily.[2] One advantage of ML is that the results are often easily interpreted and can easily be used alongside other signals, indicators, models, and analyses to provide additional/alternative insight in Positioning Analysis.

To generate the decision trees, to produce the random forest and to do the feature importance analysis in this article, the application XLSTAT is used. XLSTAT was chosen due to its ease of use, its stability and detailed help files. Furthermore, to use XLSTAT, no programming knowledge is required, and it runs within the Excel application as an add-in. Python is an excellent alternative, and arguably one of the leading programming languages in ML, but it naturally requires some programming skill.[3]

Decision Trees

The objective of this section is to provide a full explanation in a series of stages and examples, of how decision trees are constructed and how they can be used in Positioning Analysis.

A decision tree is a supervised learning algorithm designed to predict an output variable, for example the weekly return of crude oil (WTI). The output variable can also be called the target variable.

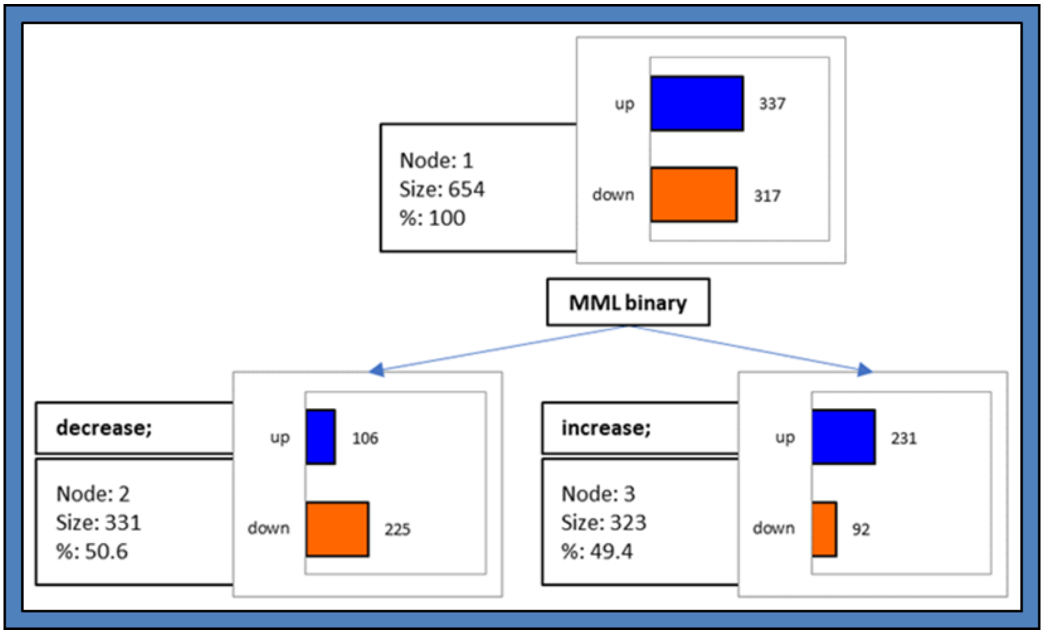
The function of a decision tree is to split the output dataset (the price returns) into two or more subsets, for example positive and negative returns, using specific decision rules derived from the input dataset, for example changes in positioning data. The variables contained in the input dataset are known as the features.

The decision tree identifies the most differentiating features in the input dataset, as well as the threshold values that best split the output dataset into the most homogeneous subsets.

Decision Trees Using Binary Data

Figure 1 on the next page gives a simple example of a binary decision tree set-up to predict whether prices of crude oil (WTI) increased or decreased over the week based on binary changes in Money Manager Long (MML) and Money Manager Short (MMS) open interest, as reported by the CFTC in the Disaggregated Commitments of Traders (COT) report every Friday.[4] The decision tree identifies the variable (either MML or MMS) that best separates the weeks where prices increased from those where prices decreased. The underlying algorithm used here only uses two binary features: whether MML or MMS increased or decreased their positions in terms of open interest.

Figure 1
Decision Tree Based on Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices)



Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The underlying data consists of 654 weeks (June 2006 to December 2018) of MM positioning data and price data. The weeks run from Tuesday to Tuesday, to be aligned with the COT release schedule and prices are therefore contemporaneous to the positioning data.

The algorithm then identifies the "best question to ask" to generate the most homologous (most uniform) child nodes. The question is: whether MMLs (as opposed to MMSs) changed their position. The underlying algorithm is explained fully in "The Decision Tree Algorithm – How Does it Work?" on the following pages.

Observations from the decision tree diagram in Figure 1:

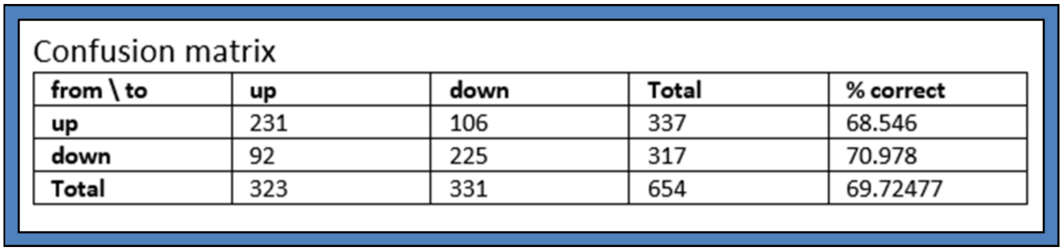
- Node 1 (also referred to as the root node) shows the 654 weeks divided into 337 weeks where crude oil (WTI) prices increased and 317 weeks where they decreased. It is therefore also called an up-node.

- Node 2 shows that out of the 654 weeks, MMLs decreased their position in 331 weeks. Out of those 331 weeks, crude oil (WTI) prices rose in 106 of the weeks (32%) and fell in 225 of the weeks (68%). It is therefore also called a down-node.

- Node 3 shows that out of the 654 weeks, MMLs increased their position in 323 weeks. Out of those 323 weeks, crude oil (WTI) prices rose in 231 of the weeks (72%) and fell in 92 of the weeks (28%).

The confusion matrix below, a table displaying the number of successful and unsuccessfully-classified observations for each of the categories shows that 69.72% of points in the entire dataset of 654 weeks were correctly classified by this tree – a good percentage. Simply put, this means that increases (decreases) in MML position are associated with up (down) moves in crude oil (WTI) prices.

Table 1
Confusion Matrix Based on Binary Input Changes (MML and MMS) and Binary Output Changes (Crude Oil (WTI) Prices)



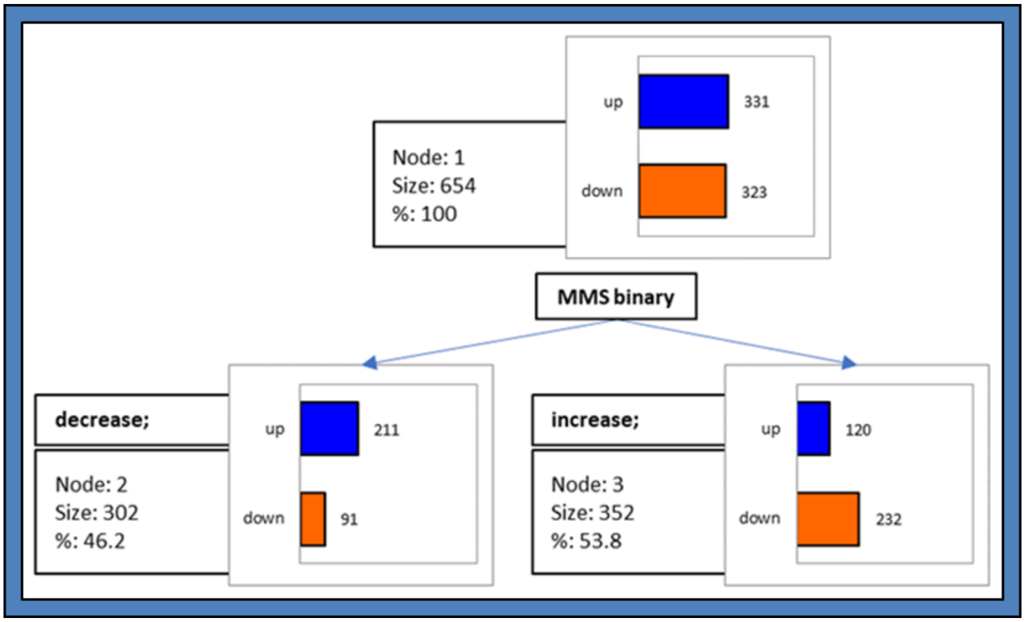
Confusion matrix

from \ to	up	down	Total	% correct
up	231	106	337	68.546
down	92	225	317	70.978
Total	323	331	654	69.72477

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Figure 2 shows a similar decision tree, but for natural gas (NG). Here the tree, set up in the same way as the tree in Figure 1 identifies the "best question" to ask as whether MMSs (as opposed to MMLs) changed their position.

Figure 2
Decision Tree Based on Binary Input Changes (MML and MMS) and Binary Output Changes (Natural Gas Prices)



Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The Decision Tree Algorithm – How Does it Work?

The decision tree in Figure 1 splits the tree according to whether MMLs changed their position, whereas for natural gas in Figure 2, it was the opposite, asking whether MMSs changed their position.

The criterion used here to split the tree is the Gini index. It is a measure of dispersion within a dataset and measures the degree of homogeneity. A value of 0 means the dataset is perfectly homogeneous, while values near 0.5 represent a heterogeneous dataset (0.5 is the highest value). In this binary example, where prices either increased or decreased, a homogeneous sample would consist solely of price increases or price decreases, whereas a heterogeneous set would consist of a mix of price increases and price decreases.

The Gini index is calculated as follows:

$$Gini = 1 - \left(\left(\frac{\text{number of price increases}}{\text{number of observations}}\right)^2 + \left(\frac{\text{number of price decreases}}{\text{number of observations}}\right)^2 \right)$$

The Gini index for the root node in Figure 1 is calculated to be:

$$Gini = 1 - \left(\left(\frac{337}{654}\right)^2 + \left(\frac{317}{654}\right)^2 \right) = 0.500$$

To then split a branch into two, all the possible Gini scores of all the possible splits are calculated. In this example there are two possible splits, by changes in MMLs or by changes in MMSs. The split that yields the two branches with the highest degree of homogeneity on average, computed as the lowest weighted average Gini score, is chosen.

For example, after a split by changes in MMLs, two branches containing 331 and 323 observations are generated. The Gini score of the left branch is 0.435 while the Gini score of the right branch is 0.407. As a result, the weighted average Gini score is:

$$Weighted\ Average\ Gini = \frac{(331 * 0.435) + (323 * 0.407)}{(331 + 323)} = 0.421$$

Splitting by changes in MMS would have generated two branches (this is not shown, but can be calculated from the underlying data), each containing 333 (left branch) and 321 (right branch) observations. The Gini score of the left branch would be 0.444, while the Gini score for the right branch would be 0.460. Here the weighted average Gini score would be:

$$Weighted\ Average\ Gini = \frac{(333 * 0.444) + (321 * 0.460)}{(333 + 321)} = 0.452$$

As the weighted-average Gini score is lower in the first case (0.421 is less than 0.452) and therefore more homogeneous, the algorithm decides to split the data by looking at MML positions.[5]

Validating the Tree

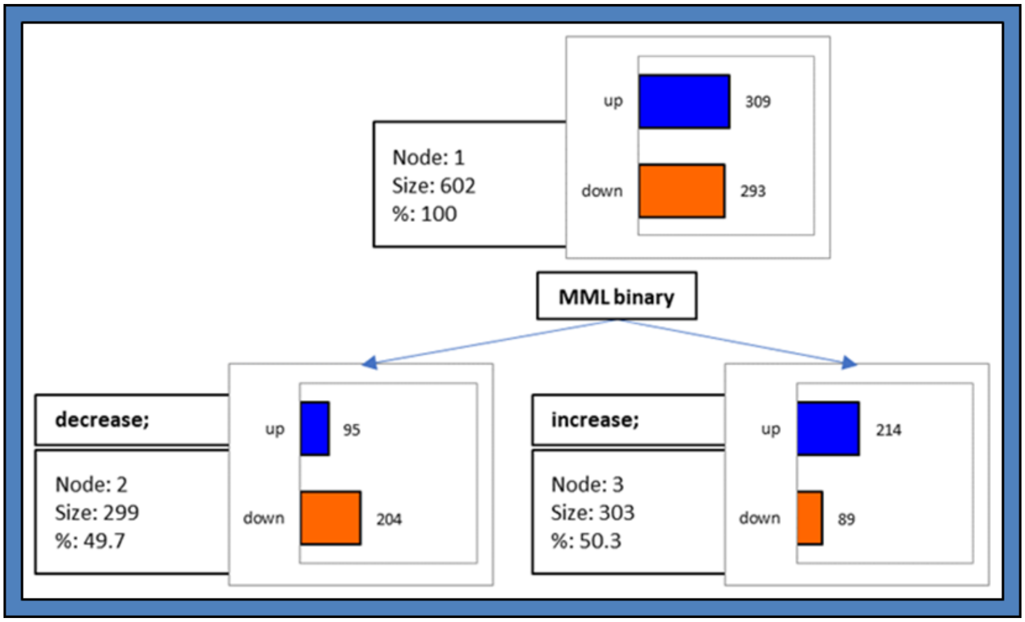
In the decision trees above, the entire dataset of 654 weeks has been used. To validate the robustness of the tree in Figure 1, it must be tested on unseen data.

The approach is to first "train" the algorithm on some portion of the data and then "test" it on another portion of data. The testing portion could, for example, be the last 52 weeks of the dataset and the training portion the 602 weeks before that.

Figure 3 shows a new decision tree trained only on the first 602 weeks of data (the training dataset).

Figure 3

Decision Tree Based on Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



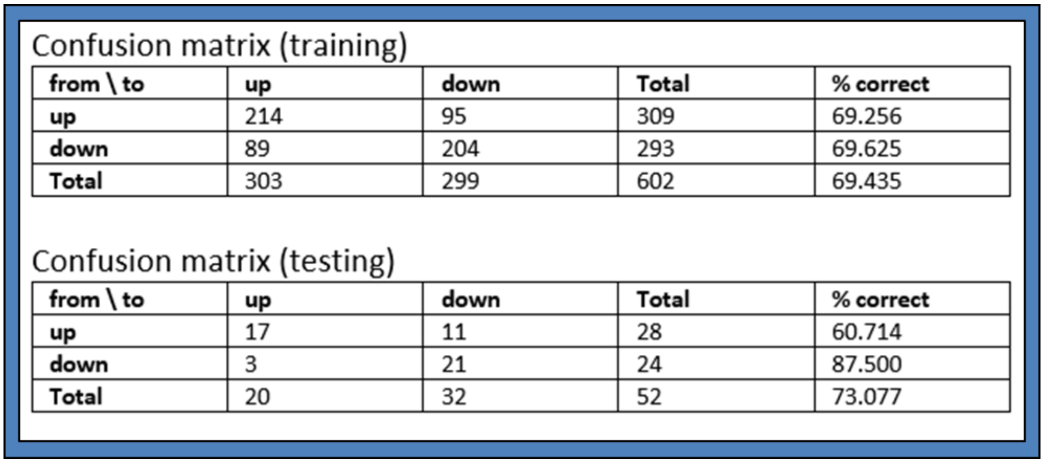
Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

This tree identifies the same question to ask first as the tree in Figure 1 – whether MMLs increased or decreased their position.

Applying the decision tree to last 52 weeks in the dataset (the testing dataset) there were 38 weeks (73%) where the algorithm correctly predicted price direction by asking whether MMLs increased or decreased their position.

When two datasets are used, two confusion matrices can now be generated: one for the training set (602 weeks) and one for the testing set (52 weeks). These are shown in Table 2 on the next page. In the training set, 69.44% of the points were correctly classified, and in the testing set, 73.10% of the points were correctly predicted – an improvement of a few percentage points.

Table 2
Confusion Matrix Based on Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
up	214	95	309	69.256
down	89	204	293	69.625
Total	303	299	602	69.435

Confusion matrix (testing)

from \ to	up	down	Total	% correct
up	17	11	28	60.714
down	3	21	24	87.500
Total	20	32	52	73.077

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

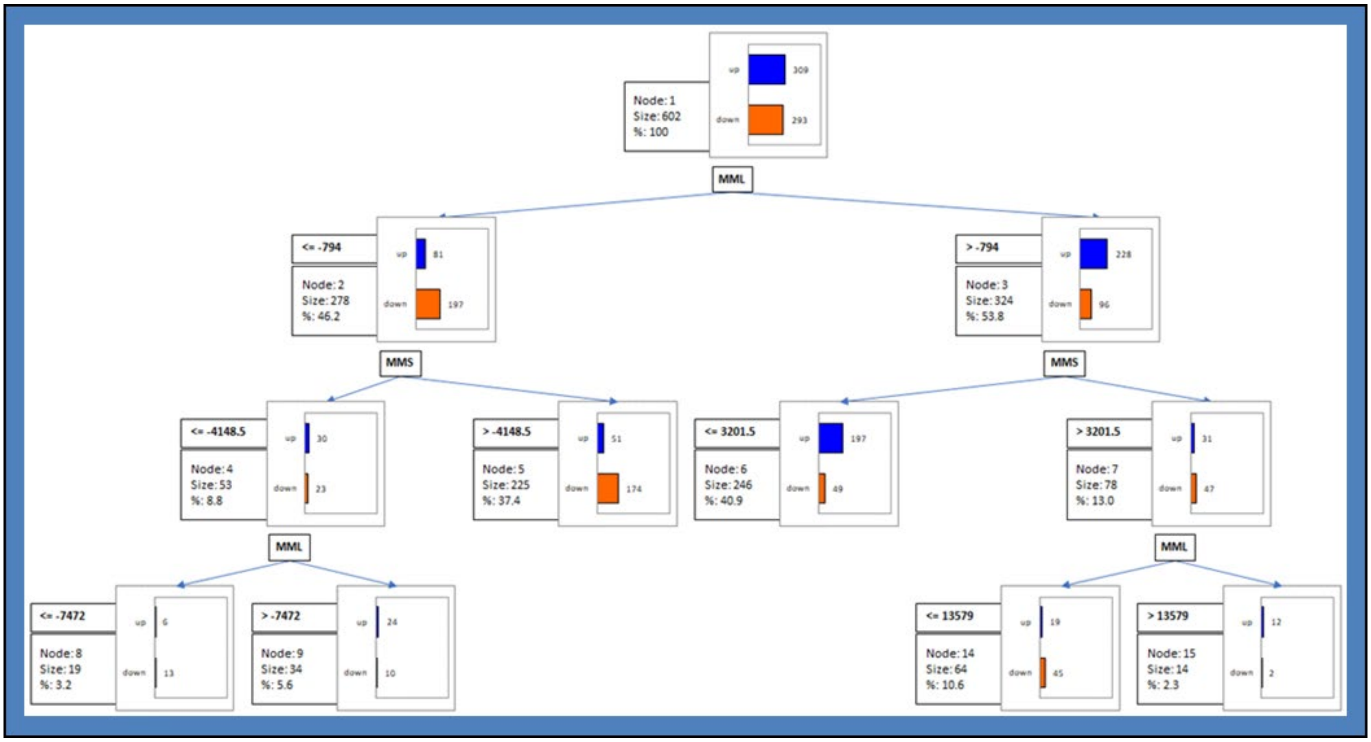
Decision Trees Using Non-Binary Data

Only binary data has been used so far – either an increase or a decrease in MM positioning and whether prices were up or down. The algorithm also works with non-binary data, deciding not only which feature to best split the tree with, but also at what threshold.

With greater possibilities, non-binary trees can grow exponentially during training as they fit the data, and ultimately, without any constraints, they will fit all the data as every scenario would be captured. If they grow too big, however, they become overfitted, and they risk performing poorly when applied to testing samples. One solution is to cut the tree before it gets too big.

Figure 4 on the next page shows the non-binary decision tree to predict whether prices of crude oil (WTI) increased or decreased over the week based on changes in MML and MMS positions over that week. The tree is cut at three levels of depth. The data has again been divided into a training set (602 weeks) and a testing set (52 weeks) as explained previously in "Validating the Tree." The first question the decision-tree asks now is whether MMs changes their position by more or less than -794 contracts.

Figure 4
Decision Tree Based on Non-Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Focusing on the first three nodes in the tree:

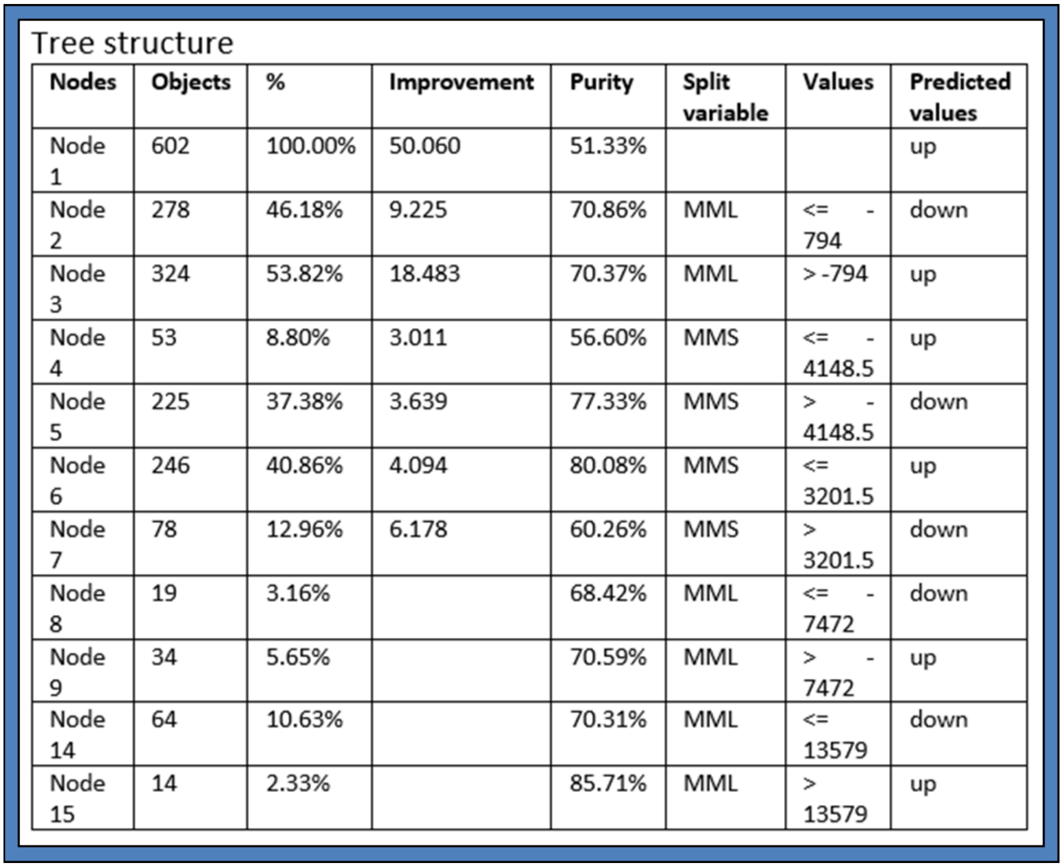
- Node 1 in Figure 4 shows the 602 weeks divided into 309 weeks where crude oil (WTI) prices increased and 293 weeks where they decreased (this is the same as Figure 3).

- Node 2 shows that out of the 602 weeks, MMLs changed their position by less than or equal to -794 contracts in 278 weeks. Out of those 278 weeks, crude oil (WTI) prices rose in 81 of the weeks and fell in 197 of the weeks.

- Node 3 shows that out of the 602 weeks, MMLs changed their position by more than -794 contracts in 324 weeks. Out of those 324 weeks, crude oil (WTI) prices rose in 228 of the weeks and fell in 96 of the weeks.

The complete tree structure for Figure 4 is shown below in Table 3. This provides a detailed description of key statistics and actions at each node in the tree.

Table 3
Tree Structure Based on Non-Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



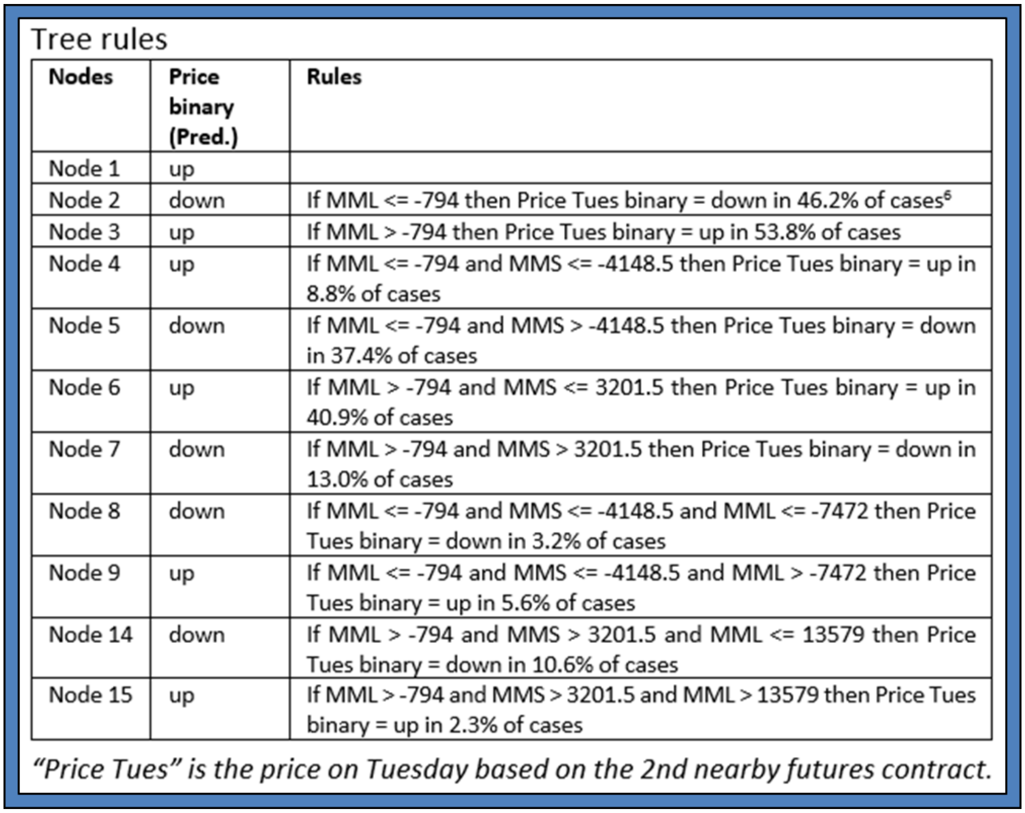
Tree structure

Nodes	Objects	%	Improvement	Purity	Split variable	Values	Predicted values
Node 1	602	100.00%	50.060	51.33%			up
Node 2	278	46.18%	9.225	70.86%	MML	<= -794	down
Node 3	324	53.82%	18.483	70.37%	MML	> -794	up
Node 4	53	8.80%	3.011	56.60%	MMS	<= -4148.5	up
Node 5	225	37.38%	3.639	77.33%	MMS	> -4148.5	down
Node 6	246	40.86%	4.094	80.08%	MMS	<= 3201.5	up
Node 7	78	12.96%	6.178	60.26%	MMS	> 3201.5	down
Node 8	19	3.16%		68.42%	MML	<= -7472	down
Node 9	34	5.65%		70.59%	MML	> -7472	up
Node 14	64	10.63%		70.31%	MML	<= 13579	down
Node 15	14	2.33%		85.71%	MML	> 13579	up

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The tree rules for Figure 4 are shown in Table 4 below. This table provides a description of the actions at each node.

Table 4
Decision Rules Based on Non-Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



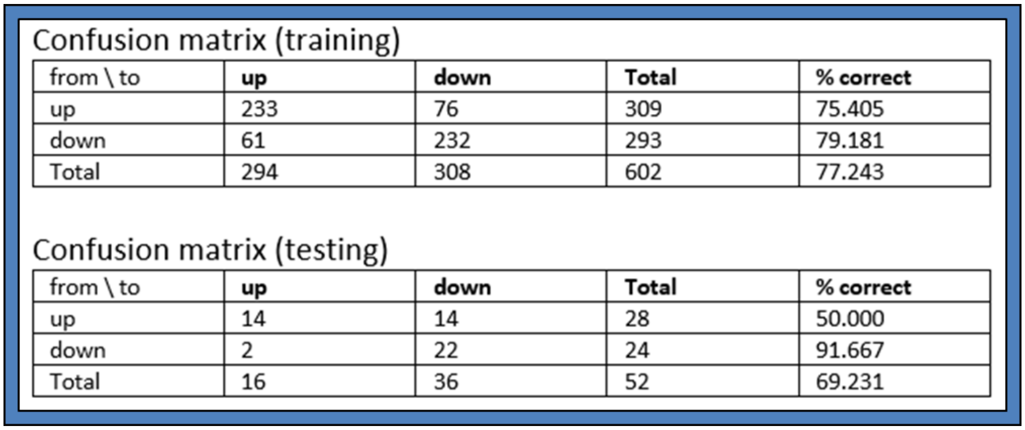
Tree rules

Nodes	Price binary (Pred.)	Rules
Node 1	up	
Node 2	down	If MML <= -794 then Price Tues binary = down in 46.2% of cases[6]
Node 3	up	If MML > -794 then Price Tues binary = up in 53.8% of cases
Node 4	up	If MML <= -794 and MMS <= -4148.5 then Price Tues binary = up in 8.8% of cases
Node 5	down	If MML <= -794 and MMS > -4148.5 then Price Tues binary = down in 37.4% of cases
Node 6	up	If MML > -794 and MMS <= 3201.5 then Price Tues binary = up in 40.9% of cases
Node 7	down	If MML > -794 and MMS > 3201.5 then Price Tues binary = down in 13.0% of cases
Node 8	down	If MML <= -794 and MMS <= -4148.5 and MML <= -7472 then Price Tues binary = down in 3.2% of cases
Node 9	up	If MML <= -794 and MMS <= -4148.5 and MML > -7472 then Price Tues binary = up in 5.6% of cases
Node 14	down	If MML > -794 and MMS > 3201.5 and MML <= 13579 then Price Tues binary = down in 10.6% of cases
Node 15	up	If MML > -794 and MMS > 3201.5 and MML > 13579 then Price Tues binary = up in 2.3% of cases

"Price Tues" is the price on Tuesday based on the 2nd nearby futures contract.

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Table 5
Confusion Matrix Based on Non-Binary Input Changes (MML and MMS) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
up	233	76	309	75.405
down	61	232	293	79.181
Total	294	308	602	77.243

Confusion matrix (testing)

from \ to	up	down	Total	% correct
up	14	14	28	50.000
down	2	22	24	91.667
Total	16	36	52	69.231

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The confusion matrices in Table 5 above show that for the training set (602 weeks) 77.24% of the points were correctly classified, and in the testing set, 69.23% of the points were correctly predicted. For the training dataset, the results are higher than the binary tree in Table 2, but a little lower for the testing dataset. This could suggest that the direction of MM activity is sufficient in predicting prices, and adding threshold information, does not add any significant incremental value.

The robustness of decision trees as an analytical tool is covered later in the section on "Random Forests."

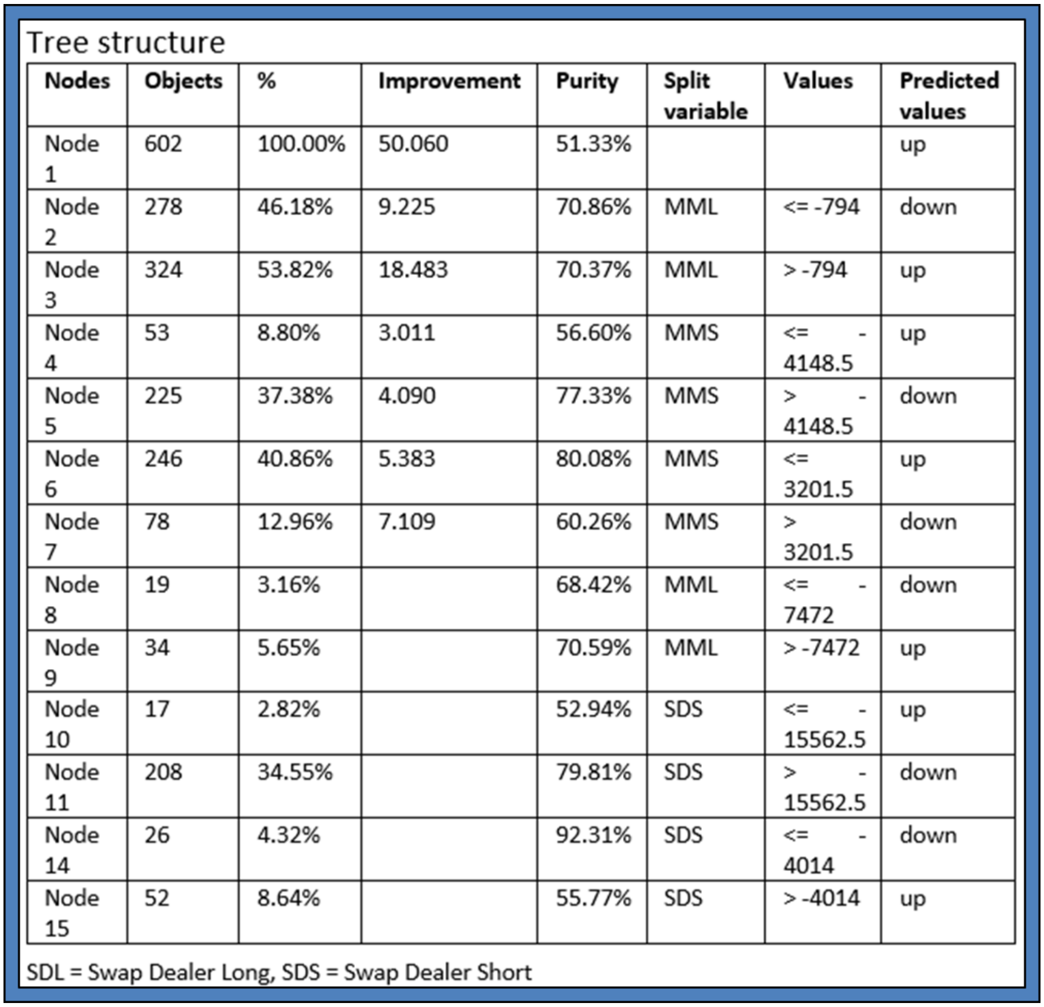
Extending the Tree to All Trader Groups

The decision tree in the section on "Decision Trees Using Non-Binary Data" uses only the MML and MMS groups as features. In the following two pages, Table 6 shows the tree structure after including all trader groups (Other Reportables (OR), Producer/Merchant/Processor/User (PMPU) and Swap Dealers (SD), with the corresponding confusion matrix shown in Table 7. The full decision tree diagram is not shown in the interest of space.

The Appendix explains the various trader groups as reported in the Disaggregated COT report.

Table 6

Tree Structure Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Tree structure

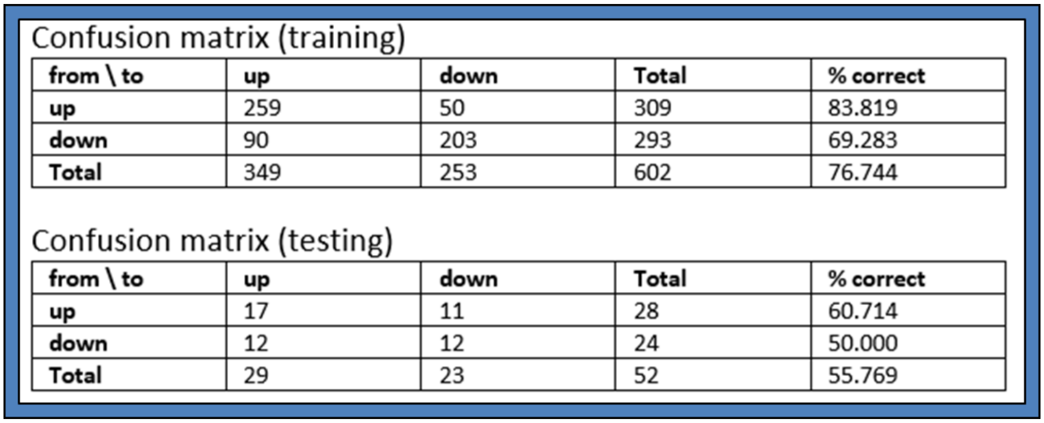
Nodes	Objects	%	Improvement	Purity	Split variable	Values	Predicted values
Node 1	602	100.00%	50.060	51.33%			up
Node 2	278	46.18%	9.225	70.86%	MML	<= -794	down
Node 3	324	53.82%	18.483	70.37%	MML	> -794	up
Node 4	53	8.80%	3.011	56.60%	MMS	<= -4148.5	up
Node 5	225	37.38%	4.090	77.33%	MMS	> -4148.5	down
Node 6	246	40.86%	5.383	80.08%	MMS	<= 3201.5	up
Node 7	78	12.96%	7.109	60.26%	MMS	> 3201.5	down
Node 8	19	3.16%		68.42%	MML	<= -7472	down
Node 9	34	5.65%		70.59%	MML	> -7472	up
Node 10	17	2.82%		52.94%	SDS	<= -15562.5	up
Node 11	208	34.55%		79.81%	SDS	> -15562.5	down
Node 14	26	4.32%		92.31%	SDS	<= -4014	down
Node 15	52	8.64%		55.77%	SDS	> -4014	up

SDL = Swap Dealer Long, SDS = Swap Dealer Short

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The initial nodes within the tree are still split along MM features, with the SD category featuring at node 10 onwards. Overall this shows that the algorithm still chooses the MM group as the most important feature in predicting price direction.

Table 7
Confusion Matrix Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (WTI Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
up	259	50	309	83.819
down	90	203	293	69.283
Total	349	253	602	76.744

Confusion matrix (testing)

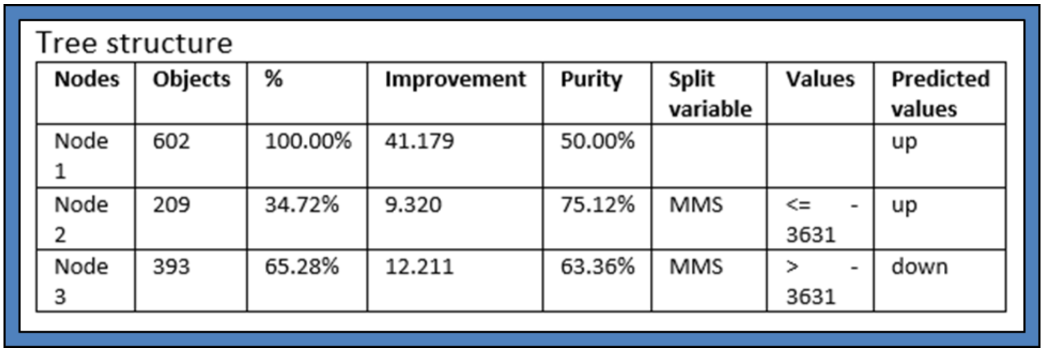
from \ to	up	down	Total	% correct
up	17	11	28	60.714
down	12	12	24	50.000
Total	29	23	52	55.769

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Whilst the algorithm shows the MM group to be the most important feature, it is also important to understand that these results do not infer causality – especially when the data are set up in a contemporaneous way (as in all the examples so far.) Just because the trees mostly identify the MM group as an important feature in predicting prices, it does not mean that MMs are driving prices. It could easily mean that MMs are following prices.

Looking at natural gas and copper, a similar profile emerges with the MM group continuing to be the most important feature. Table 8 shows the tree structure, and Table 9 on the next page documents the confusion matrix for all trader groups for natural gas.

Table 8
Tree structure Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (Natural Gas Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks

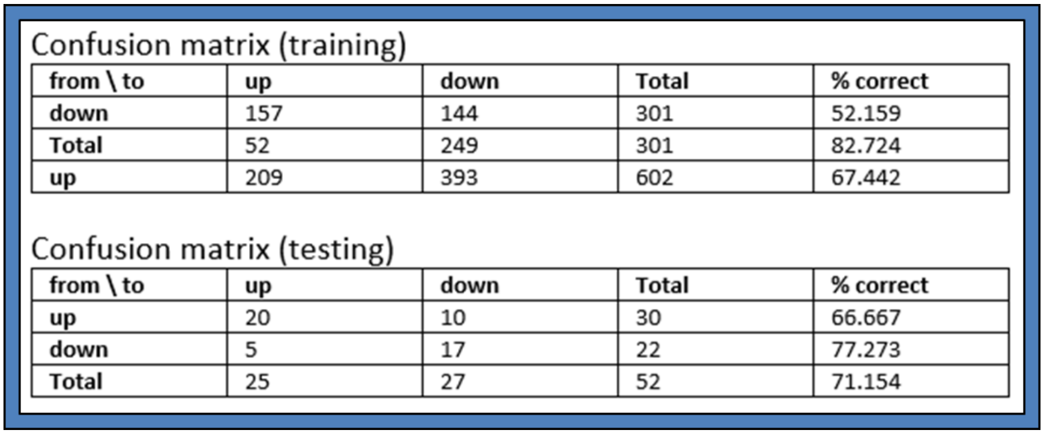


Tree structure

Nodes	Objects	%	Improvement	Purity	Split variable	Values	Predicted values
Node 1	602	100.00%	41.179	50.00%			up
Node 2	209	34.72%	9.320	75.12%	MMS	<= - 3631	up
Node 3	393	65.28%	12.211	63.36%	MMS	> - 3631	down

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Table 9
Confusion Matrix Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (Natural Gas Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
down	157	144	301	52.159
Total	52	249	301	82.724
up	209	393	602	67.442

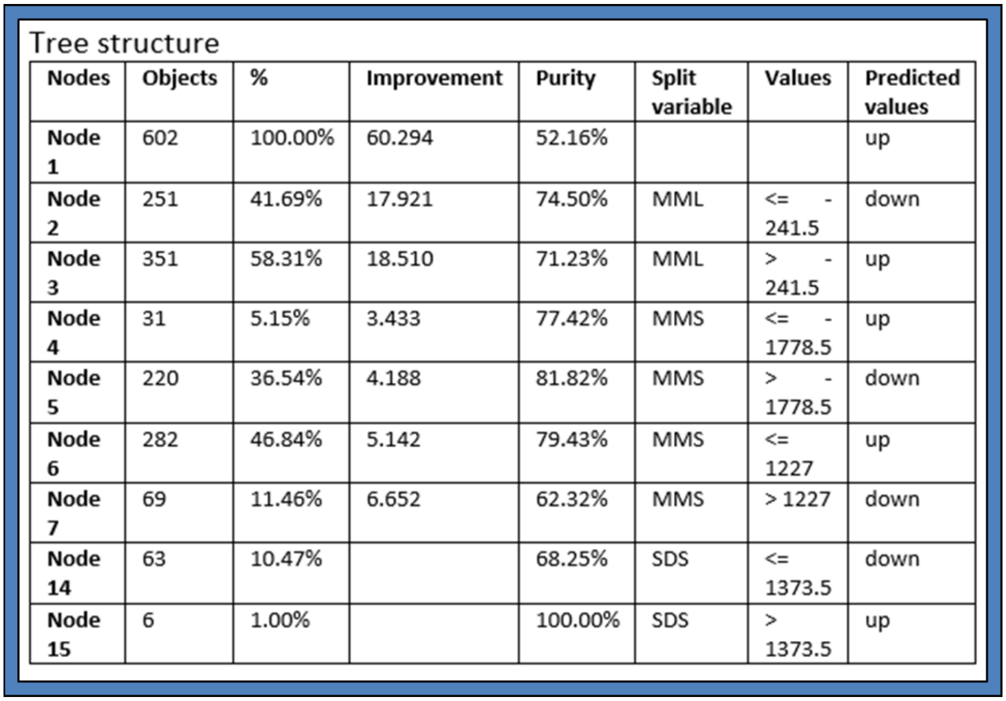
Confusion matrix (testing)

from \ to	up	down	Total	% correct
up	20	10	30	66.667
down	5	17	22	77.273
Total	25	27	52	71.154

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Table 10 shows the tree structure, and Table 11 on the next page provides the confusion matrix for all trader groups for copper.

Table 10
Tree Structure Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (Copper Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks

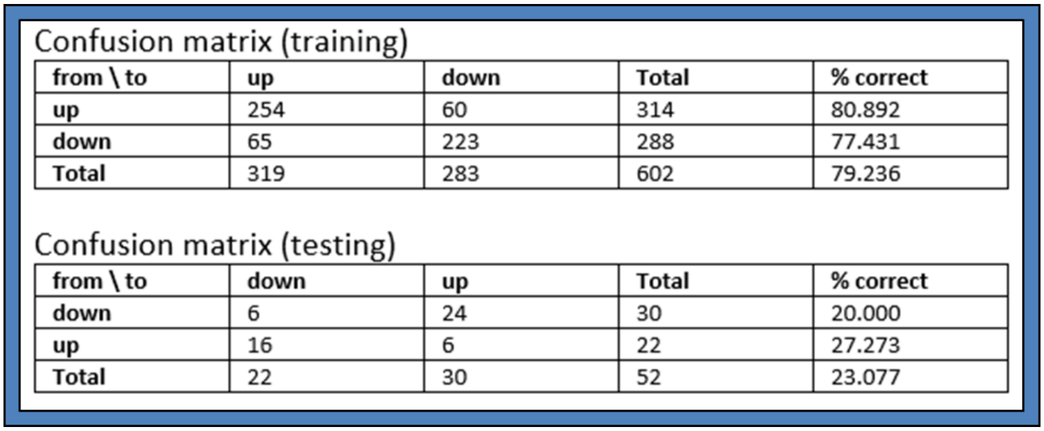


Tree structure

Nodes	Objects	%	Improvement	Purity	Split variable	Values	Predicted values
Node 1	602	100.00%	60.294	52.16%			up
Node 2	251	41.69%	17.921	74.50%	MML	<= -241.5	down
Node 3	351	58.31%	18.510	71.23%	MML	> -241.5	up
Node 4	31	5.15%	3.433	77.42%	MMS	<= -1778.5	up
Node 5	220	36.54%	4.188	81.82%	MMS	> -1778.5	down
Node 6	282	46.84%	5.142	79.43%	MMS	<= 1227	up
Node 7	69	11.46%	6.652	62.32%	MMS	> 1227	down
Node 14	63	10.47%		68.25%	SDS	<= 1373.5	down
Node 15	6	1.00%		100.00%	SDS	> 1373.5	up

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Table 11
Confusion Matrix Based on Non-Binary Input Changes (All Groups) and Binary Output Changes (Copper Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
up	254	60	314	80.892
down	65	223	288	77.431
Total	319	283	602	79.236

Confusion matrix (testing)

from \ to	down	up	Total	% correct
down	6	24	30	20.000
up	16	6	22	27.273
Total	22	30	52	23.077

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

As mentioned above, the robustness of decision trees as an analytical tool is covered later in the section on "Random Forests."

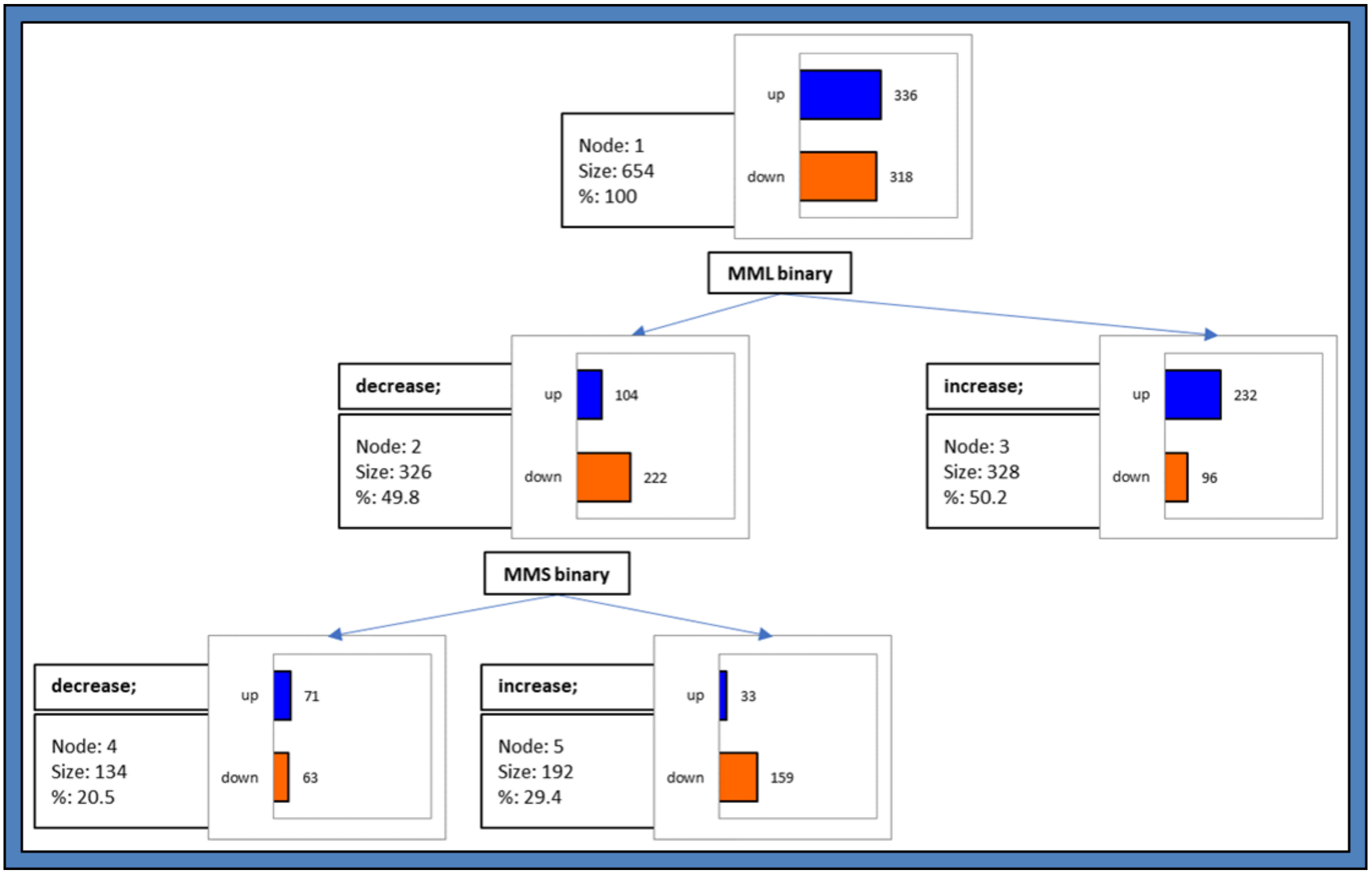
Feature Importance

An essential attribute of decision trees, by virtue of their underlying algorithm, is their ability to identify the most important features when predicting the target variable. This is called feature importance, and in the context of Positioning Analysis refers to identifying the trader groups whose changes best explain price direction.

Feature importance can be calculated by looking at the decrease in the average Gini score, the increase in homogeneity, of all the nodes split along a given feature, weighted by the number of observations in those nodes.

Figure 5 is a binary decision tree to predict whether prices of copper increased or decreased over the week based on absolute changes in MML and MMS positions over that week. The decision tree initially identifies the variable MML at the first level and followed by the variable MMS at the second level.

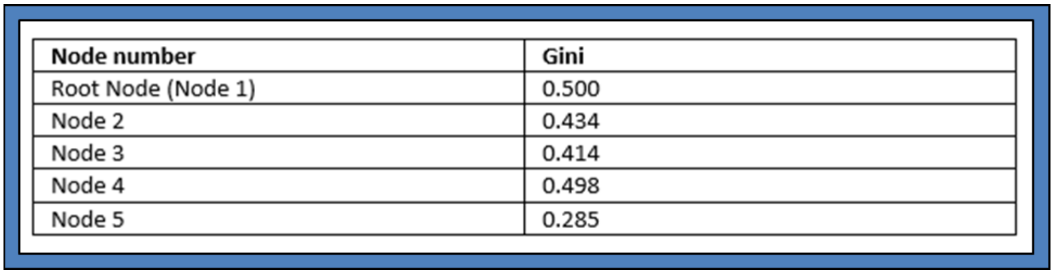
Figure 5
Decision Tree Based on Binary Input Changes (MML and MMS) and Binary Output Changes (Copper Prices)



Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

The Gini score at each of the five nodes in Figure 5, calculated previously in the section on "The Decision Tree Algorithm – How Does it Work?" are:

Table 12
Gini Scores at:



Node number	Gini
Root Node (Node 1)	0.500
Node 2	0.434
Node 3	0.414
Node 4	0.498
Node 5	0.285

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

At the first level of the tree, where the tree is split along the MML feature, the weighted average Gini score is:

$$Weighted\ Average\ Gini = \frac{(326 * 0.434) + (328 * 0.414)}{(326 + 328)} = 0.424$$

The decrease in Gini at this level is = 0.500 – 0.424 = 0.076. This is attributable to the MML feature.

At the second level, where the tree is split along the MMS feature, the weighted average Gini score is:

$$Weighted\ Average\ Gini = \frac{(134 * 0.498) + (192 * 0.285)}{(134 + 192)} = 0.373$$

The decrease in Gini at this level is = 0.424 – 0.373= 0.051. This is attributable to the MMS feature.

The individual feature importance is then calculated as:

$$Feature\ importance_{MML} = \frac{0.076}{0.076 + 0.051} = 60\%$$

$$Feature\ importance_{MML} = \frac{0.051}{0.076 + 0.051} = 40\%$$

Random Forests

By way of a summary – in the previous section on "Decision Trees," single decision trees were used to classify specific datasets. To verify the robustness, the dataset was split into a training set (602 weeks) and a testing set (52 weeks). The objective of the training set was to calibrate the model before then testing it on the testing set. Confusion matrices then report the success of the predictions in each of these datasets.

The problem with this approach is there is the risk of overfitting, making the robustness questionable. To address overfitting issues and to increase the robustness of the decision tree approach, random forests are a satisfactory solution.

The Random Forest Methodology

In a random forest many different decision trees are generated. Each tree is trained on a random subset of the training dataset also using a subset of the available features.

Each of these trees is then applied to the testing data and its prediction of the target variable made. Each prediction is then averaged using a voting mechanism within the algorithm across all the trees.

For example, a random forest could be "grown" as follows:

- 100 different subsets from the training set (with replacement) are created, each containing 100 weeks of data.[7] Each sample can include intermittent (i.e. non-consecutive) points, and because the sampling is done with replacement, a given observation may appear several times in any given sample.

- 100 different trees are then generated, each of which is trained on one of the 100 subsets. Instead of using the full set of eight features however, each tree only uses *n* randomly selected features. Using the square root of the total number of features is widely considered to be a good value for *n*. This means that for a total set of eight features, each tree would use three randomly selected features.

- Once all the trees are trained, the forest is then applied to the testing data with each tree in the forest making its own independent prediction. The final results are then voted upon. If for example, 75 trees predict a price increase, and 25 predict a price decrease, the forest is then said to predict a price increase.

The usefulness of random forests, above and beyond that of decision trees, lies in the robustness of the approach and the ability to overcome overfitting. By having many different (random) trees generate independent predictions based on different subsets of the features, the variance in predictions is reduced and much of the risk in overfitting reduced. Random forests rely on the "wisdom of crowds": individual trees (as described throughout the section on "Decision Trees") can make classification mistakes, but on average, the whole forest will make more robust and accurate predictions.

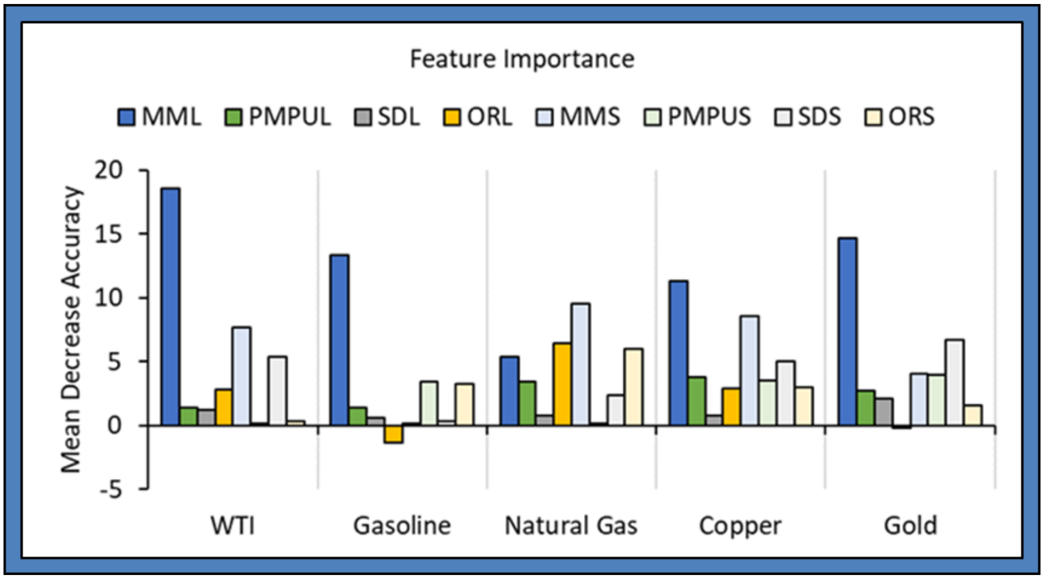
Random Forest Feature Importance

In the same way that feature importance is calculated for a single decision tree, the feature importance from a random forest can also be calculated. The advantage is that the results are also more robust. The feature importance from a random forest provides meaningful insight into the trader group changes that best explain price changes.

Two different measures of importance are given for each feature in the random forest. The first is based on the decrease of Gini score as described in the section on "Feature Importance." The second, called the Mean Decrease Accuracy, is based on how much the accuracy decreases when a variable is excluded.[8] The XLSTAT application uses the second method.

Figures 6 and 7 on the next page show the feature importance profiles for 10 major commodities for the entire dataset between 2006 and 2018.

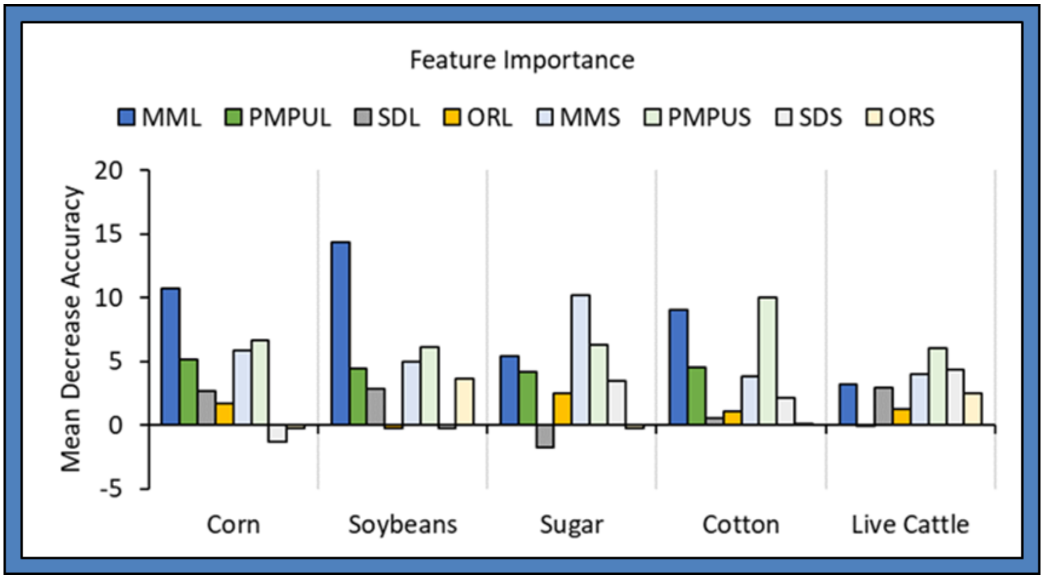
Figure 6
Feature Importance (i)



Source: Based on data from Bloomberg.

Producer/Merchant/Processor/User (PMPU), Swap Dealer (SD), Money Manager (MM), Other Reportables (OR). The L or S suffix further splits between Long and Short.

Figure 7
Feature Importance (ii)



Source: Based on data from Bloomberg.

Producer/Merchant/Processor/User (PMPU), Swap Dealer (SD), Money Manager (MM), Other Reportables (OR). The L or S suffix further splits between Long and Short.

For crude oil (WTI), gasoline, gold and soybeans, the MML group is the most important, whereas for natural gas and sugar it is the MMS group. For cotton and live cattle, the PMPUS group is the most important.

Dynamic Feature Important and Alternative Features

By using a rolling window approach, or an anchored walk forward approach, the evolution of feature importance can be tracked over time as new data becomes available. This provides insight into shifts and changes in the market structure or in trader behavior.

The variables included as features have so far have been focused only on changes in the directional (long and short) open interest of trader groups, but spreading data could also be included. The number of traders can also be used, either as an exclusive set of variables or in combination with the open interest variables.

Feature importance can be a compelling way of identifying new patterns and relationships in positioning that can help improve how other positioning signals, indicators, and models are interpreted and used.

Using ML to Trade

In each of the sections in this article, the price (output) variable has always been contemporaneous to the positioning (input) variables as the objective of the approach has been entirely analytically driven. ML has been used to uncover relationships between changes in positioning and changes in commodity prices - specifically, as described at the outset of the piece, which aspects of positioning are most useful in helping to understand commodity markets from a machine-learning perspective.

Having the price variables contemporaneous to the positioning variables means the model is not directly tradeable. Naturally the insights revealed in "Random Forest Feature Importance," for example, can be used to enhance trading strategies based on positioning data, but the results have not been generated from a tradeable framework.

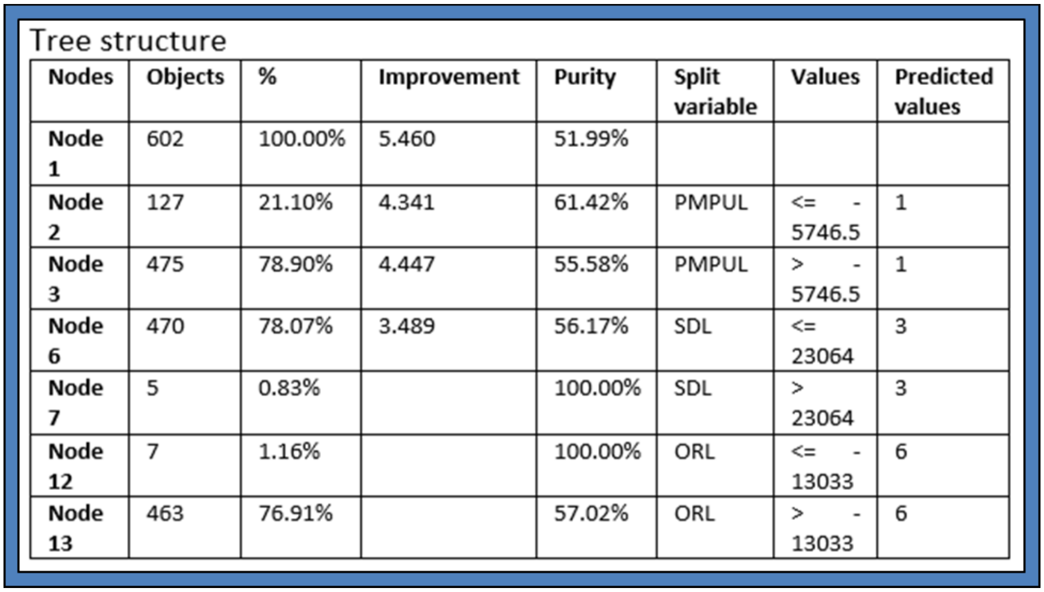
The data release schedule of COT positioning data is fully explained on the CFTC website, but to summarize, positioning data is released every Friday (with all CFTC related data after the market close and ICE COT data just before the market close) and refers to the previous Tuesday.[9] Changes in positioning in this article run from Tuesday to Tuesday and the price changes have been set to the same schedule also.

To make the relationship tradeable, the price information in the decision tree needs to be changed from Tuesday to Tuesday to the following Monday to Monday. This is because the COT data is released on Friday, mostly after the market closes, and the earliest opportunity to trade is Monday.

Tables 13 and 14 on the next page are like the tree structure in Table 8 and confusion matrix in Table 9, except the decision tree now predicts whether prices of natural gas increased or decreased in the following week, running from Monday to Monday, but still based on changes in positioning from the previous week. Simply, when the positioning data is released on the Friday (for the previous Tuesday) the algorithm predicts whether prices will rise (fall) in the upcoming Monday to Monday period. In having shifted the price data, this now makes the model "tradeable", allowing any trades to be placed on the close on Monday night.[10]

Table 13

Tree Structure Based on Binary Input Changes (All Groups) and Absolute Output Changes (Natural Gas Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



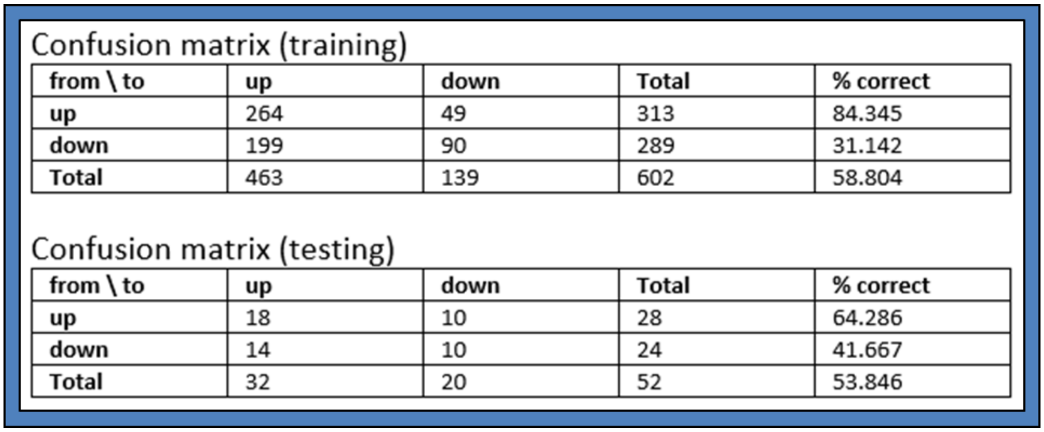
Tree structure

Nodes	Objects	%	Improvement	Purity	Split variable	Values	Predicted values
Node 1	602	100.00%	5.460	51.99%			
Node 2	127	21.10%	4.341	61.42%	PMPUL	<= - 5746.5	1
Node 3	475	78.90%	4.447	55.58%	PMPUL	> - 5746.5	1
Node 6	470	78.07%	3.489	56.17%	SDL	<= 23064	3
Node 7	5	0.83%		100.00%	SDL	> 23064	3
Node 12	7	1.16%		100.00%	ORL	<= - 13033	6
Node 13	463	76.91%		57.02%	ORL	> - 13033	6

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Table 14

Confusion Matrix Based on Binary Input Changes (All Groups) and Absolute Output Changes (Natural Gas Prices) – Training Dataset 602 Weeks, Testing Dataset 52 Weeks



Confusion matrix (training)

from \ to	up	down	Total	% correct
up	264	49	313	84.345
down	199	90	289	31.142
Total	463	139	602	58.804

Confusion matrix (testing)

from \ to	up	down	Total	% correct
up	18	10	28	64.286
down	14	10	24	41.667
Total	32	20	52	53.846

Source: Based on data from Bloomberg. Output taken directly from XLSTAT.

Interestingly, the tree structure in Table 13 shows the addition of four more nodes due to having shifted the price data. Different features throughout the tree are also seen compared to Table 8 which highlights the significance of the change.

The confusion matrix in Table 14 shows a large deterioration in performance, due to the lag in the price data. Table 14 shows 58.80% of points in the training dataset are classified correctly but falling to 53.84% for the testing dataset. Similar deteriorations are seen for other commodities and at this level the success of the model is close to being random.

Extending this single tree to a random forest framework also does not improve the results. Importantly, this suggests that ML as a trading tool based on positioning data used in this way is unsatisfactory. The requirement to lag the price data to make the model tradeable, reduces the performance of the results significantly.

It is important to understand that the performance of ML in a tradable framework in no way renders ML an ineffective tool for analysis.

On the contrary, ML can be highly effective in identifying new patterns and relationships in the data that would be extremely challenging to identify through other channels. As mentioned above, ML can help improve how other positioning signals, indicators, and models are interpreted and used.

Appendix

Disaggregated COT Report Categories

Producer/Merchant/Processor/User (PMPU)
A "producer/merchant/processor/user" is an entity that predominantly engages in the production, processing, packing or handling of a physical commodity and uses the futures markets to manage or hedge risks associated with those activities.

Swap Dealer (SD)
A "swap dealer" is an entity that deals primarily in swaps for a commodity and uses the futures markets to manage or hedge the risk associated with those swap transactions. The swap dealer's counterparties may be speculative traders, like hedge funds, or traditional commercial clients that are managing risk arising from their dealings in the physical commodity.

Money Manager (MM)
A "money manager," for the purpose of this report, is a registered commodity trading advisor (CTA); a registered commodity pool operator (CPO); or an unregistered fund identified by the CFTC. These traders are engaged in managing and conducting organized futures trading on behalf of clients.

Other Reportables
Every other reportable trader that is not placed into one of the other three categories is placed into the "other reportable" category.

Source: https://www.cftc.gov/MarketReports/CommitmentsofTraders/index.htm

Endnotes

This article is excerpted from Chapter 14 of *Advanced Positioning, Flow and Sentiment Analysis in Commodity Markets*, which was published by Wiley in January 2020.

1 Decision trees where the target variable is discrete are called classification trees. Decision trees where the target variable can take continuous values are called regression trees.

2 Non-linear relationships are where the change in one entity does not correspond with constant change in another entity.

3 https://www.xlstat.com/en/

4 Please see the Appendix for a description of each trader category.

5 Besides the Gini index, other decision criteria exist, including some based-on information theory (entropy) and intra-group variance. Only the Gini index is used in the article.

6 The percentage cases represent the proportion of the sample size at that node.

7 Any number for samples can be taken of any length.

8 The importance measure used in the XLSTAT application for a given variable is the mean error increase of a tree when the observed values of this variable are randomly exchanged in the OOB (Out-Of-Bag) samples.

For each tree, the prediction error on the out-of-bag data is computed. Then the same is done after permuting each explanatory variable. The difference between the two is then averaged over all trees, and according to the choice of the user, normalized or not by the standard deviation of the differences. If the standard deviation of the differences is equal to 0 for a variable, the division is not done.

In classification, in addition to the impact of permutations on the overall error of the forest, we also measure the impact on each of the modalities of the response variable. Source: XLSTAT help files (https://www.xlstat.com/)

9 https://www.cftc.gov/MarketReports/CommitmentsofTraders/index.htm

10 The trades can be placed at any time on a Monday, but the final settlement price of the day is used as this is tradeable (via futures executed at TAS (Trade at Settlement) or via S&P GSCI or BCOM excess return indices) and no estimates for slippage effectively need to be factored into account. Whilst the model has been trained to predict prices over the full week Monday to Monday, a trade can naturally be exited earlier, and the model can also be configured for shorter price periods – for example over a single day.

Author Biography

MARK KEENAN
Head of Research and Strategy at Engelhart Commodity Trading Partners; and Editorial Advisory Board Member, *Global Commodities Applied Research Digest*

Mr. Mark Keenan is Head of Research and Strategy at Engelhart Commodity Trading Partners (ECTP) and previously Managing Director, Global Commodities Strategist, and Head of Research for Asia-Pacific at Société Générale Corporate & Investment Bank (SG CIB). He has over 20 years of experience in commodity quantitative analysis, research and strategy across all the major energy, metal, agriculture and soft commodities markets.

Author of *Advanced Positioning, Flow and Sentiment Analysis in Commodity Markets*, published by Wiley and also *Positioning Analysis in Commodities Markets – Bridging Fundamental and Technical Analysis*, Mr. Keenan appears regularly on CNBC and

Bloomberg television and is quoted widely in global press and media channels. He has a Master's degree in Molecular and Cellular Biochemistry from Oxford University.

Mr. Keenan previously provided two expert analyses for the *GCARD*. He contributed to the Summer 2018 issue of the *GCARD* where he described positioning analysis in the commodity markets; and he also co-authored an article for the Winter 2018 issue on cryptocurrencies, Bitcoin and blockchain.