Machine Learning in Financial Research: Exploring Drivers for Commodity Future Prices

Mathis Makarski, *Naoki Yamamoto Laboratory*

***Abstract*—This paper explores the intersection of computer science and finance, focusing on applying machine learning methods and signal processing techniques to uncover the factors influencing price changes in traded securities. This research focuses on commodity futures, leveraging data sources such as the Commissions of Traders Report by the CFTC to construct a feature space. It aims to build a supervised learning problem and employ ensemble learning techniques, such as decision tree-based bagging classifier, to model and explain price changes in commodity futures. Additionally, we will address critical details involved in applying machine learning methods to financial problems. This includes techniques like triple barrier labeling, feature importance analysis, systematic feature elimination and applying information metrics. Results show that using the proposed data sources to construct a dicision-tree based classifier do not result in a model that is superior to a random classifier.**

***Index Terms*—Decision Tree Classifier, Commodities, Future Contracts**

I. Introduction

THE integration of computer science and machine learning methods is rapidly gaining momentum in today's financial landscape, fueled by the increasing availability of computational resources. This paradigm shift is particularly relevant in addressing complex challenges in the financial industry, particularly the development of profitable trading models, which requires extensive research efforts. Central to this research effort are the cutting-edge methodologies proposed by De Prado in his books [1], [2], which provide advanced tools and frameworks that serve as the basis for our approach in this paper.

The complexity of financial markets requires a nuanced understanding of the characteristics and underlying market forces that drive price movements. To address this complex landscape, we employ methods such as the decision tree bagging classifier, triple barrier labeling, feature importance methods, and information metrics. These techniques, grounded in De Prado's framework, enable our research to unravel the subtleties inherent in financial data, opening the way for informed financial modeling and building performant trading models.

Our focus is on constructing a classifier that predicts movements in gold futures prices using a variety of fundamental data sources as features, since the futures market characterized by its high liquidity has a rich set of recorded data. The futures market, characterized by its centralized exchange and voluminous trading contracts, provides not only an exemplary problem for our research, but also an opportunity to explore and refine computational models.

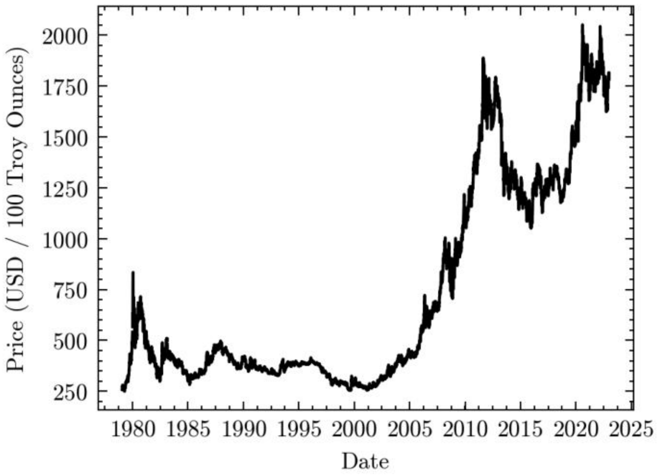
This paper is structured to comprehensively address the challenges of constructing a classifier in the field of finance. The subsequent Data section outlines the process of constructing a coherent future price series, as well as the use of the Commitment of Traders (COT) report, the application of an oscillation filter, and the incorporation of a price momentum indicator, leading to the construction of a well-defined feature space. The model section details the use of a decision tree bagging classifier, while the feature selection section details the application of permutation feature importance and systematic elimination to refine the feature set. Results and Critical Discussion follow, providing insight into the performance of the model and engaging in analytical discourse. Finally, the Conclusion section summarizes the key findings and outlines avenues for future research.

II. Data

Divided into Commodity Futures Prices, Commitment of Traders Report, Oscillation Filter, Differencing and Standard Scaler, each subsection shows the steps of data pre-processing and representation that are critical to the subsequent modeling and analysis phases.

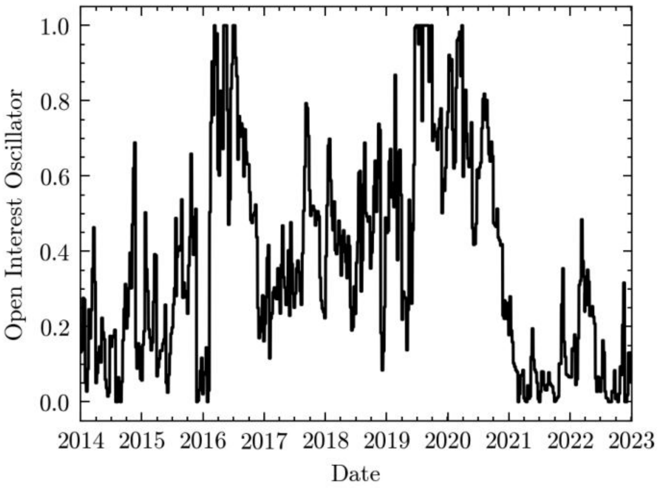
A. Commodity Future Prices

Commodity futures are contracts that represent agreements to buy or sell a specific commodity at a predetermined price on a future date. Multiple contracts exist for the same underlying commodity, each corresponding to a different expiration month, and are actively traded on the exchange at the same time. This research focuses on a specific commodity, the gold future (GC), with contract months spanning Feb, Apr, Jun, Aug, Oct, and Dec. An important consideration in dealing with commodity futures data is the management of contract expirations. When a contract expires, the next year's contract for the same expiration month begins trading. In order to construct a coherent single future time series for algorithm training, the study adopts the "nearest future continuation" approach. Applying this approach on GC results in the price series, shown in figure 1. In the nearest future method, the next month's contract is concatenated to the price series whenever the previous month's contract expires. While the nearest future continuation method accurately reflects historical price levels, it does have a limitation. It does not accurately reflect changes in equity because it requires traders to roll positions - sell an expiring contract and buy the next month's contract. An alternative approach, known as continuous spread-adjusted future continuation, addresses this concern [3]. However, for



This figure shows a coherent gold price series, derived by using the nearest future continuation method.

Fig. 1. Gold Nearest Future Continuation



This figure shows an excerpt of the oscillation filter applied on the open interest time series.

Fig. 2. Open Interest Oscillator

the sake of simplicity, this research chooses to adhere to the ”nearest future” method and neglect the changes in equity resulting from the rolling process.

B. Commitment of Traders Report

The Commitment of Traders (COT) report, which is publicly available on the Commodity Futures Trading Commission (CFTC) website, serves as a resource that provides weekly insights into the positioning of various categories of traders. These include commercials (Comm), non-commercials (Non-Comm) or institutional traders, and non-reportables (Non-Rept). Commercials, often representing corporations, use futures to hedge their exposure to commodity price fluctuations related to their production or consumption. In contrast, institutional investors typically take speculative positions, often opposing those of the commercials. Non-reportable traders include small traders who trade below the reportable volume limit. These categories include both long and short positions, as well as open interest, which represents the total number of contracts traded. Recognizing that this report reveals fundamental aspects of market structure, this paper seeks to determine whether such insights can be leveraged for trading advantage. Consequently, these time series derived from the COT report form the first part of the feature space in this paper.

C. Oscillation Filter

The Stochastic Oscillator is used by traders to generate signals that indicate overbought or oversold market conditions. It measures the momentum of a given time series and provides insight into trends and potential reversals. [4] The Oscillator Value OSC_k at a discrete point in time k is calculated using the following formula, including the value of the underlying time series p_k, the high $H_{k,n}$, and respectively the low $L_{k,n}$, within the past n days:

$$OSC_k = \frac{p_k - L_{k,n}}{H_{k,n} - L_{k,n}} \tag{1}$$

This filter moves within the range of 0 to 1, ensuring a standardized output. In this research, the oscillation filter is applied to all time series extracted from the COT report, using a lookback window of $n = 730$ days, which is an arbitrarily chosen value, as signals from the COT report are expected to be long-term in nature. As an example the oscillation filter applied on the open interest time series is included and can be retrieved from figure 2.

D. Moving Average Indicator

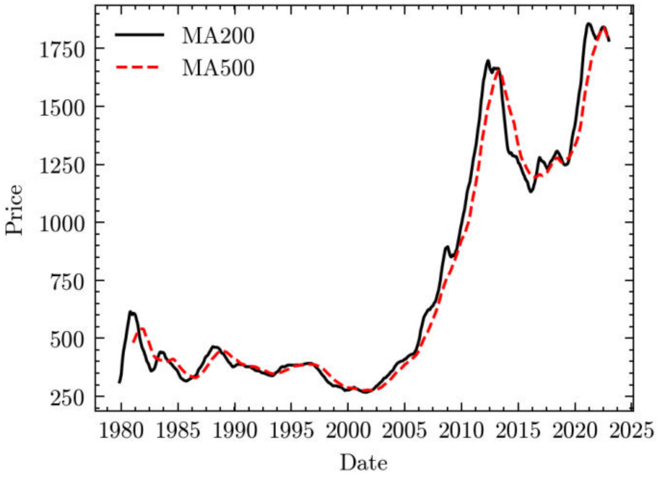
The Moving Average indicator is used to model price momentum by taking the difference between two simples moving averages (MA) of the same underlying price series, each calculated with different rolling window sizes. In the first step, the moving average with rolling window size n is calculated as follows:

$$MA_k = \frac{1}{k} \sum_{i=n-k+1}^{n} p_i \tag{2}$$

To build the indicators in this study, MA100 (moving average with window size $n = 100$ days) is subtracted from MA200, which represents the short term price momentum indicator. As a long term price momentum indicator, MA200 is substracted from MA500, both are shown in figure 3.

E. Feature Space Analysis

Before using our feature space in a learning algorithm, it is important to consider potential interplay between our features. Features that share similar information can induce substitution effects that affect the results. For example, in methods such as permutation feature importance that will be used later, if two nearly identical features exist, their importance may be halved



This figure shows both moving averages used to compute the long term price momentum indicator. The underlying time series from which the moving average derived is the gold price future series.

Fig. 3. Simple Moving Averages

due to equal selection probability, leading to underestimation [2]. To mitigate such effects and to ensure a robust analysis, we perform a careful feature preselection, eliminating the simultaneous use of highly correlated features. We assess shared information through a two-part evaluation: a correlation matrix analysis and an examination of information metric variation.

The correlation matrix, shown in Figure 4, is calculated by the standard correlation coefficient, also known as Pearson correlation coefficient, between two features for each combination and provides a visual representation of the relationships between the feature time series, which are sorted to reveal blocks of high correlation. Although two blocks show up in the correlation matrix, which show values between ±0.5, we argue that these are not significantly high enough to indicate substitution effects.

In scenarios where nonlinearity is prevalent, the variation of information turns out to be a more appropriate distance metric than the correlation measure. [2] This metric allows us to address questions about the unique information provided by a random variable, all without imposing specific functional assumptions. Consider X a discrete random variable that takes a value x from the set S_X with probability $p[x]$. The entropy of X is defined as

$$H[X] = -\sum_{x \in S_X} p[x] log[p[x]] \tag{3}$$

The joint entropy of X and Y is

$$H[X,Y] = -\sum_{x,y \in S_X \times S_Y} p[x,y] log[p[x,y]] \tag{4}$$

The conditional entropy is defined as

$$H[X|Y] = H[X,Y] - H[Y] \tag{5}$$

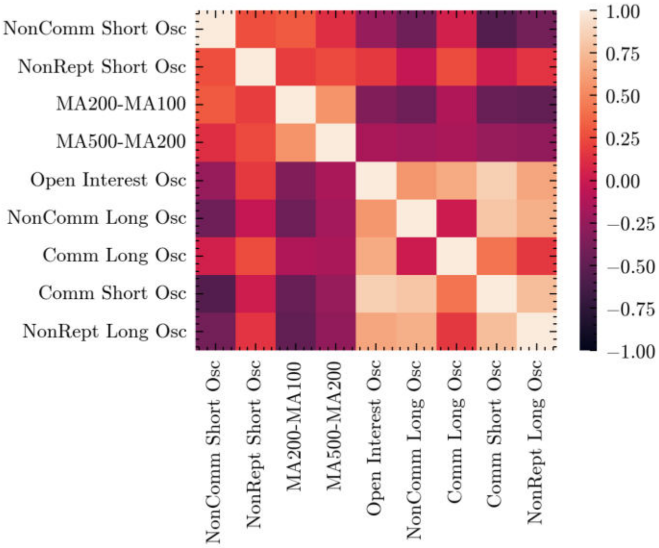


Fig. 4. Correlation Matrix

The mutual information is defined as the decrease in uncertainty in X that results from knowing the value of Y:

$$I[X,Y] = H[X] - H[X|Y] = H[X] + H[Y] - H[X,Y] \tag{6}$$

Now we can define and transform the equation of the variation of information to

$$VI[X,Y] = H[X|Y] + H[Y|X] = H[X,Y] - I[X,Y] \tag{7}$$

By standardizing the variation of information

$$\tilde{VI}[X,Y] = \frac{VI[X,Y]}{H[X,Y]} = 1 - \frac{I[X,Y]}{H[X,Y]} \tag{8}$$

we bound it between zero and one, thus making it compareable. Further, variation of information is a metric because it satisfies nonnegativity, symmetry $VI[X,Y] = VI[Y,X]$, and the triangle inequality. The variation of information measure can be interpreted as the uncertainty we expect in one variable when we know the value of the other. So, a lower value indicates that more information is shared between both variables. As seen in figure 5, all values are are higher then approximately 0.8, thus one can argue that all features can be used without the possibility of substitution effects.

III. Model

This section describes the approach used to construct a predictive framework through supervised learning, employing the decision tree bagging classifier as proposed in [1]. To facilitate supervised learning, a critical prerequisite is the derivation of labels from the future price series. Following this, the model section discusses the choice of the decision tree bagging classifier and shows its implementation in the context of our research framework.

A. Triple Barrier Labeling

The triple barrier labeling technique, introduced in [1], is particularly relevant to our research because of its significant

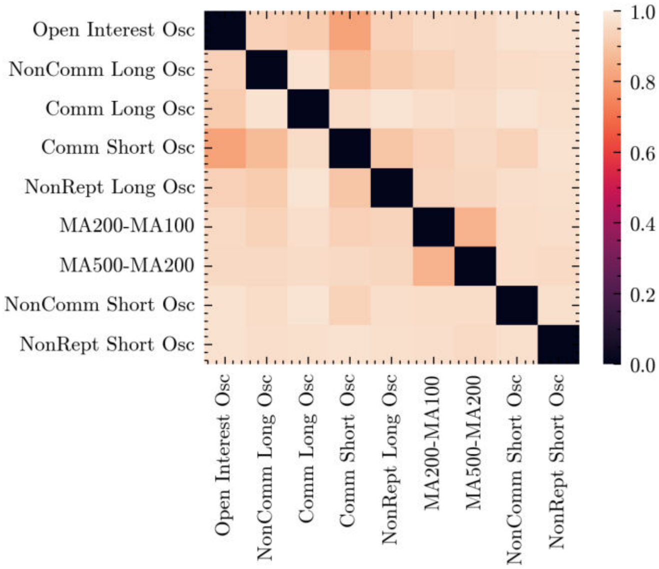


Fig. 5. Variation of Information Matrix

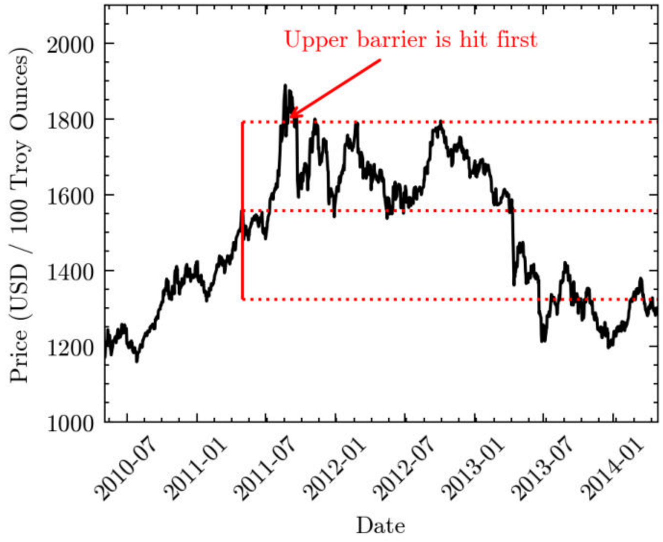


Fig. 6. Triple Barrier Labeling

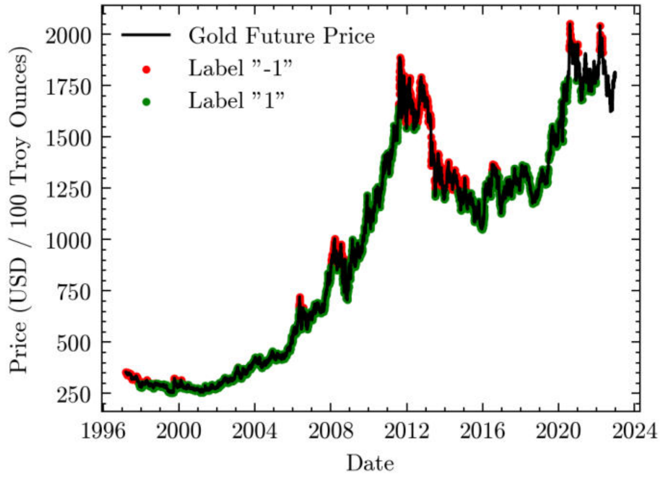


Fig. 7. Derived Labels

similarity to the perspective of an investment professional, representing a fundamental difference from the typical forecasting problem in the prevalent literature. The three barriers include two horizontal and one vertical barrier, each serving a specific purpose in the labeling process. The profit-taking and stop-loss targets dynamically define the horizontal barriers, and the vertical time barrier represents an expiration limit based on the number of bars that have passed since initiating the position. If the upper barrier is touched first, the observation is labeled ”1”. If the lower barrier is touched first, the observation is labeled ”-1”. If the vertical barrier is touched first, we have two options: Label the observation based on the sign of the outcome, reflecting realized profit or loss, or label the observation as ”0,” indicating that the position resulted in neither profit nor loss within the defined limits. Figure 6 shows an example how one data point will be labeled according to the triple barrier labeling technique.

In this paper, the profit-taking and stop-loss targets are $\pm 15\%$, which is an arbitrarily chosen value. Furthermore, for the sake of simplicity, no vertical time barrier is used. Employing these parameters, figure 7 shows the accordingly derived labels.

B. Decision Tree Bagging Classifier

The model employed in this paper is constructed using a combination of the decision tree and the bagging classifier from the well-known sklearn library, as shown in the listing below. This approach is similar to Random Forests, although it has distinct characteristics and advantages. Further insights into their different strengths and properties are elaborated in chapter 6.4 of [1].

Listing 1. Model Implementation

```
from sklearn.tree import
    DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
clf = DecisionTreeClassifier(criterion='
    entropy', max_features=1, class_weight='
    balanced', min_weight_fraction_leaf=0)
clf = BaggingClassifier(estimator=clf,
    n_estimators=1000,max_features=1,
    max_samples=avgU, oob_score=False)
```

In scenarios where a significant portion of the samples have non-identically and independently distributed (non-IID) characteristics, the problem of overfitting can persist. This is due to the process of sampling with replacement, which results in the construction of a significant number of essentially identical decision trees. Unfortunately, this overfitting phenomenon is a well-known weakness of decision trees, as each tree tends to capture noise and irregularities in the data. In [1], solutions for specific parameter settings are provided to counteract this problem. An essential technique for dealing with non-IID characteristics is to set the `max_samples` parameter of the bagging classifier to the average label uniqueness. This adjustment is critical because our label values are derived from different points in time, spanning different time intervals that overlap. A detailed description of this approach can be found in chapter 4 of [1], explaining how adjusting the `max_samples` parameter to label uniqueness effectively counteracts non-IID characteristics.

IV. Feature Selection

Based on the feature space derived in the Data section, the primary goal of the Feature Selection section is to identify the subset of features that optimally contribute to the construction of a performant classifier. The method used for this evaluation is Permutation Feature Importance, which assesses the importance of each feature within the classifier model. A systematic elimination process is then applied to refine the feature set, ensuring that the retained features collectively maximize the performance of the classifier.

A. Permutation Feature Importance

Permutation Feature Importance (PFI) is a method for assessing the importance of individual features within a supervised learning model. The procedure begins by fitting the bagging classifier, which serves as a baseline model, to the data set. In this section, we use cross-entropy loss as the relevant performance metric. The essential step in PFI is to isolate each feature individually and shuffle its values, effectively creating a randomized version of that particular feature while keeping all other features intact. With this modified data set, the bagging classifier is retrained to reflect its performance when the original relationship between the feature and the target variable is disrupted. By comparing the cross-entropy loss of the new model L_1 to the baseline model L_0, we calculate the feature importance value F as the difference in cross-entropy loss

$$F = L_0 - L_1 \tag{9}$$

which quantifies the impact of the shuffled feature on the model's predictive power. Thus, a positive feature importance value means a drop in model performance after shuffling, indicating that the shuffled feature contains important information. It is important to note that using the 10-fold purged cross-validation procedure, we obtain 10 feature importance values per feature from which we derive mean and standard deviation. The shuffling process is repeated for each feature in each cross-validation step.

B. Systematic Feature Elimination

Starting with the initial feature space as presented in the Data section, our systematic feature elimination process uses permutation feature importance to measure the importance of each feature within the classifier model. Iteratively, the feature with the smallest negative importance is eliminated in each cycle. This process continues until all features have importance values greater than zero. It is important to note that while this method refines the feature set to improve model performance, it does not exhaustively explore all possible subsets of features within the initial feature space. Given the computational cost of evaluating every combination, our systematic elimination strikes a balance between efficiency and model optimization.

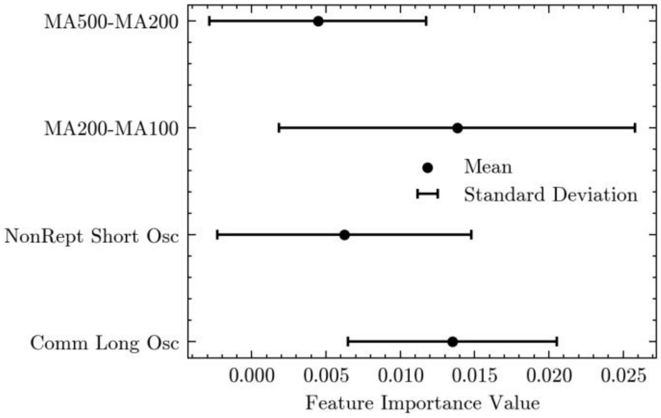


Fig. 8. Feature Importance of the optimized Model

V. Results and Discussion

After applying the feature selection methods described in the previous section, the refined feature set includes MA500-MA200, MA200-MA100, NonRept Short Osc, Comm Long Osc. The feature importance means and standard deviations are visually presented in Figure 8, where MA200-MA100 and Comm Long Osc have the highest feature importance, although each feature is accompanied by a large standard deviation. After rerunning the model with the optimized feature set, we evaluate the accuracy for each split within the cross-validation procedure. The results show an average accuracy of 0.7684. This value seems high at the first look, nevertheless it is not outstanding, since the labels derived by the triple barrier labeling method consist of approximately 77% positive labels. Thus a model that randomly picks 50% of the time a positive label would still have an accuracy of 77% and the presented model in this paper is not significantly better than a random label picking model.

VI. Conclusion

In conclusion, this paper has undertaken a comprehensive exploration of predictive modeling in the financial domain, encompassing the construction of a classifier. The data section begins with the construction of a coherent future price series, integrating the Commitment of Traders (COT) report, an oscillation filter, and a price momentum indicator to form a well-defined feature space. Using a decision tree bagging classifier, we navigated the complexities of modeling financial data. The feature selection approach, using permutation feature importance and systematic elimination, aimed to refine a set of features for optimal model performance while keeping computational cost low.

Despite achieving a notable accuracy value, it is important to recognize that the model's performance, does not exceed that of a random classifier, although it is high. This discrepancy suggests avenues for future research. Exploring alternative data sources for features could reveal new insights, while investigating the explainability and economic implications of our results adds depth to the interpretability of the model's predictions. Furthermore, extending the application of our approach to other commodities can provide valuable comparative insights and contribute to the broader landscape of predictive modeling in financial markets.

REFERENCES

[1] de Prado, M. L., *Advances in Financial Machine Learning*, Wiley, 2018

[2] de Prado, M. L., *Machine Learning for Asset Managers*, Cambridge University Press, 2020

[3] Schwager, J. D., *A Complete Guide to the Futures Market*, Wiley, 2017

[4] Pruitt G., *The Ultimate Algorithmic Trading System Toolbox + Website: Using Today's Technology To Help You Become A Better Trader*, Wiley, 2016