Is the Covid equity bubble rational? A machine learning answer

Jean Jacques Ohana [1] **Eric Benhamou** [2 3] **David Saltiel** [2 4] **Beatrice Guez** [2]

Abstract

Is the Covid Equity bubble rational? In 2020, stock prices ballooned with S&P 500 gaining 16%, and the tech-heavy Nasdaq soaring to 43%, while fundamentals deteriorated with decreasing GDP forecasts, shrinking sales and revenues estimates and higher government deficits. To answer this fundamental question, with little bias as possible, we explore a gradient boosting decision trees (GBDT) approach that enables us to crunch numerous variables and let the data speak. We define a crisis regime to identify specific downturns in stock markets and normal rising equity markets. We test our approach and report improved accuracy of GBDT over other ML methods. Thanks to Shapley values, we are able to identify most important features, making this current work innovative and a suitable answer to the justification of current equity level.

1. Introduction

Recent stock market news are filled more than ever with euphoric and frenetic stories about easy and quick money. Take for instance the Tesla stock. Its market capitalization is now hovering around $ 850 billion USD, despite a price earning over 134 to be compared with the automotive industry PE of 18 and a market capitalization larger than the sum of the next nine largest automotive companies. Likewise, bitcoin has gone ballistic reaching 40 000 USD despite its pure virtual status, its implicit connection to criminal crime or fraudulent money and recurrent stories of electronic wallets lost or hack. Likewise, unknown companies have been suddenly put on the front stage because of some tweets or other social media hot news. This has been for instance the case of the company Signal that was confused with the social media company and whose share exploded despite being non related to the social media Signal app. For a large group of traditional investment managers and in particular whose qualified as value, this excitement is not very understandable neither explainable or audible. Current valuations are absurd and totally disconnected from the fundamentals of the company and the macroeconomic context. In contrast, for the young millennials and in particular the *Covid generation* also referred to as the *Robinhood* traders, named after the stock exchange platform created in the United States on which all (young) Americans who are interested in stock markets are, this is completely logic. Stocks can only go up. For them, there is no need to be interested in macroeconomics to invest in the stock market. They just have to ride the wave and in particular the news and jump on stocks that everyone is looking after. For sure, markets cannot reach the sky. But at least on the short term, one must concede that markets are still in bull regime. And the real question is not whether this bubble is going to burst but when thanks to a proper understanding of the rational behind it. If we are precisely able to get the logic, we can have a chance to detect when stocks will reverse. Hence the true question is not to say if market are overvalued or not but find the key drivers and rational behind current level. To answer this question, we take a machine learning point of view as we want to use a large quantity of data and be able to extract information.

The novelty of this approach is to answer an economical debate whether equity market current valuation makes sense using a very modern and scientific approach thanks to machine learning that is able to exploit many variables and provides answer without any specific bias. In contrast to statistical approaches that are often geared towards validating a human intuition, machine learning can provide new reasoning and unknown and non linear relationship between variables and output. In this work, we explore a gradient boosting decision trees (GBDT) approach to provide a suitable and explainable answer to the rationality of Covid equity bubble.

1.1. Related works

Our work can be related to the ever growing theme of using machine learning in financial markets. Indeed, with increasing competition and pace in the financial markets, robust

[1]Homa Capital, France [2]Ai for Alpha, France [3]Lamsade, Paris Dauphine, France [4]Lisic, ULCO, France. Correspondence to: Jean Jacques Ohana <jjohana@homacapital.fr>.

forecasting methods has become a vital subject for asset managers. The promise of machine learning algorithms to offer a way to find and model non-linearity in time series has attracted lot of attention and efforts that can be traced back as early as the late 2000's where machine learning started to pick up. Instead of listing the large amount of works, we will refer readers to various works that reviewed the existing literature in chronological order.

In 2009, (Atsalakis & Valavanis, 2009) surveyed already more than 100 related published articles using neural and neuro-fuzzy techniques derived and applied to forecast stock markets, or discussing classifications of financial market data and forecasting methods. In 2010, (Li & Ma, 2010) gave a survey on the application of artificial neural networks in forecasting financial market prices, including exchange rates, stock prices, and financial crisis prediction as well as option pricing. And the stream of machine learning was not only based on neural network but also generic and evolutionary algorithms as reviewed in (Aguilar-Rivera et al., 2015).

More recently, (Xing et al., 2018) reviewed the application of cutting-edge NLP techniques for financial forecasting, using text from financial news or twitters. (Rundo et al., 2019) covered the wider topic of usage of machine learning techniques, including deep learning, to financial portfolio allocation and optimization systems. (Nti et al., 2019) focused on the usage of support vector machine and artificial neural networks to forecast prices and regimes based on fundamental and technical analysis. Later on, (Shah et al., 2019) discussed some of the challenges and research opportunities, including issues for algorithmic trading, back testing and live testing on single stocks and more generally prediction in financial market. Finally, (Sezer et al., 2019) reviewed not only deep learning methods but also other machine learning methods to forecast financial times. As the hype has been recently mostly on deep learning, it is not a surprise that most of their reviewed works are on deep learning. The only work cited that is gradient boosted decision tree is (Krauss et al., 2017)

In addition, there are works that aim to review the best algorithms for predicting financial markets. With only a few exceptions, these papers argue that deep networks outperform traditional machine learning techniques, like support vector machine or logistic regression. There is however the notable exception of (Ballings et al., 2015) that argue that Random Forest is the best algorithm when compared with peers like Support Vector Machines, Kernel Factory, AdaBoost, Neural Networks, K-Nearest Neighbors and Logistic Regression. Indeed, for high frequency trading and a large amount of input data types coming from financial news and twitter, it comes at no surprise that deep learning is the method of choice as it can incorporate large amount of input data types and in particular text inputs. But when it comes to small data set like daily data with properly formatted data from times series, the real choice of the best machine learning is not so obvious.

Interestingly, Gradient boosting decision trees (GBDT) are almost non-existent in the financial market forecasting literature. One can argue that GBDT are well known to suffer from over fitting when tackling regression problems. However, they are the method of choice for classification problems as reported by the machine learning platform Kaggle. In finance, the only space where GBDT are really cited in the literature is the credit scoring and retail banking. For instance, (Brown & Mues, 2012) or (Marceau et al., 2019) reported that GBDT are the best ML method for this specific task as they can cope with limited amount of data and very imbalanced classes.

If we are interested in classifying stock market into two regimes: a normal rising one and a crisis one, we are precisely facing very imbalanced classes and a binary classification challenge. In addition, if we are looking at daily observations, we have also a machine learning problem with limited number of data. This two points can hinder seriously the performance of deep learning algorithms that are well known to be data greedy. The interest of this approach is to have two simple regimes and avoid changes in trends that are hard to detect (Benhamou, 2018). Hence, our work has consisted in researching whether GBDT can provide a suitable method to identify regimes in stock markets. In addition, as a byproduct, GBDT provide explicit rules (even if they can be quite complex) as opposed to deep learning making it an ideal candidate to investigate regime qualification for stock markets. In this work, we apply our methodology to the US S&P 500 future. Naturally, this can be easily transposed and extended to other main stock markets like the Nasdaq, the Eurostoxx, the FTSE, the Nikkei or the MSCI Emerging future.

1.2. Contribution

Our contributions are threefold:

- we specify a valid methodology using GBDT to determine regimes in financial markets, based on a combination of more than 150 features including financial metrics, macro economics, risk aversion, price and technical indicators. Not only does this provide a suitable explanation for current equity levels thanks to features analysis but it also provides a tool to attempt for early signals should a turn point in the market come.
- we discuss in greater details technical subtleties for imbalance data sets and features selection that is key for the success of this methods. We show that for many other machine learning algorithm, selecting fewer very

specific features provides improvement across all methods.

- Finally, we compare this methodology with other machine learning (ML) methods and report improved accuracy of GBDT over other ML methods on the S & P 500 future.

1.3. Why machine learning?

The aim and promise of Machine learning (ML) is to use data without any preconceived bias, with a rigorous and scientific approach. Compared to statistics that is geared towards validating or rejecting a test, ML uses blindly the data to find relationship between them and the targeted answer. In case of supervised learning, this means finding relationship between our 150 variables and the labeled regime.

1.4. Why GBDT?

The motivations for Gradient boosting decision trees (GBDT) are multiple:

- GBDT are well know methods to provide state of the art ML methods for small data sets and classification problems. They are supposed to perform better than their state of the art brother, Deep Learning methods, for small data sets. In particular, GBDT methods have been one of the preferred methods from Kagglers and have won multiple challenges.

- GBDT methods can handle data without any prior re-scaling as opposed to logistic regression or any penalized methods. Hence they are less sensitive to data re-scaling

- they can cope with imbalanced data sets as detailed in section 3.3.

- when using the leaf-wise use leaf-wise tree growth compared to level-wise tree growth, they provide very fast training.

2. Methodology

In a normal regime, equity markets are rising as investors are paid for their risks. This has been referred to as the equity premium in the financial economics literature (Mehra & Prescott, 1985). However, there are subsequent down turns when financial markets are in panic and falling. Hence, we can simply assume that there are two regimes for equity markets:

- a *normal* regime where an asset manager should be long to benefit from the long bias of equity markets.
- and a *crisis* regime, where an asset manager should either reduce its equity exposure or even sell short it if the strategy is a long short one.

We formally say that we are in crisis regime if returns are below the historical 5 percentile computed on the training data set. The parameter 5 is not taken randomly but has been validated historically to provide meaningful levels, indicative of real panic and more importantly forecastable. For instance for the S&P 500 market, typical levels are returns at minus 6 to minus 5 percents over a period of 15 days. To make our prediction whether the coming 15 days return will be below 5 percentile (hence be classified as in crisis regime), we use more than 150 features described later on as they deserve a full description. Simply speaking these 150 features are variables ranging from implied volatility of equities, currencies, commodities, credit and VIX forward curve, to financial metrics indicators like 12 month forward estimates for sales, earning per share, price earning, macro economics surprise indexes (like the aggregated Citigroup index that compiles and Z-scores most important economic difference for major figures like ISM numbers, non farm payrolls, unemployment rates, etc).

We are looking explicitly at only two regimes with a specific focus on tailed events on the returns distribution because we found that it is easier to characterize extreme returns than to predict returns using our set of financial features. In machine learning language, our regime detection problem is a pure supervised learning exercise, with two classes classification. Hence the probability of being in the normal regime is precisely the opposite of the crisis regime probability.

In the rest, we assume daily price data are denoted by P_t. The return over a period p is simply given by the corresponding percentage change over the period: $R_t^d = P_t/P_{t-d} - 1$. The crisis regime is determined by the subset of events where returns are lower or equal to the historical 5 percentile or centile denoted by C. Returns that are below this threshold are labeled 1 while the label value for the normal regime is set to 0. Using traditional binary classification formalism, we denote the training data $X = \{x_i\}_i = 1^N$ with $x_i \in \mathbb{R}^D$ and their corresponding labels $Y = \{y_i\}_{i=1}^N$ with $y_i \in 0, 1$. The goal of our classification is to find the best *classification* function $F^*(x)$ according to the sum of some specific loss function $\mathcal{L}(y_i, F(x_i))$ as follows:

$$F^* = \underset{F}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(y_i, F(x_i))$$

Gradient boosting considers the function estimation of F to be in additive form where T is the number of boosted rounds:

$$F(x) = \sum_{m=1}^{T} f_m(x)$$

where T is the number of iterations. The set of weak learners $f_m(x)$ are designed in an incremental fashion. At the m-th stage, the newly added function, f_m is chosen to optimize the aggregated loss while keeping the previous found weak learners $\{f_j\}_{j=1}^{m-1}$ fixed. Each function f_m belongs to a set of parameterized base learners that are modeled as decision trees. Hence, in GBDT, there is an obvious design choice between taking a large number of boosted round and very simple based decision trees or a limited number of base learners but of large size. In other words, we can decide to use a small boosted round and a large decision trees whose complexity is mostly driven by its maximum depth or we can alternatively choose a large boosted round and very simple decision trees. In our experience, it is better to take small decision trees to avoid over-fitting and an important number of boosted round. In our experiment, we use 500 boosted rounds. The intuition between this design choice is to prefer a large crowd of experts that can not memorize data and hence should not over fit compared to a small number of strong experts that are represented by large decision trees. If these trees go wrong, their failure is not averaged as opposed to the first solution. Typical implementations of GBDT are XGBoost as presented in (Chen & Guestrin, 2016), LightGBM as presented (Ke et al., 2017), or Catboost as presented (Prokhorenkova et al., 2018). We tested both XGBoost and LightGBM and found an improvement in terms of speed of three time faster for LighGBM compared to XGBoost for similar learning performances. Hence, in the rest of the paper, whenever we will be mentioning GBDT, it will be indeed LightGBM.

To make experiments, we take daily historical returns for the S&P 500 merged back-adjusted future using Homa internal market data. Our daily observations are from 01Jan2003 to 15Jan2021. We split our data into three subsets:

- a train data set from 01Jan2003 to 31Dec2018
- a validation data set used to find best hyper-parameters from 01Jan2019 to 31Dec2019
- and a test data set from 01Jan2020 to 15 Jan2021

2.1. GBDT hyperparamers

GBDT have a lot of hyper parameters to specify. To our experience, the following hyper parameters are very relevant for imbalanced data sets and need to be fine tuned using evolutionary optimisations as presented in (Benhamou et al., 2019c), (Benhamou et al., 2019b), (Benhamou et al., 2020a) or for its discrete version in (Benhamou et al., 2019a)

- min sum hessian in leaf
- min gain to split
- feature fraction
- bagging fraction
- lambda l2

There is a parameter playing a central role in the proper use of GBDT which is the max depth. On the S&P 500 future, we found that very small trees with a max depth of one performs better over time than any larger tree. These 5 parameters mentioned above are determined as the best hyper parameters on the validation set.

2.2. Features used

As we can see in figure 1, the model is fed by more than 150 features to derive a daily 'crash' probability. These data can be grouped into 6 families:

- **Risk aversion metrics** such as implied volatility of equities, currencies or commodities.
- **Price indicators** such as returns or equity-bond correlation.
- **Financial metrics** such as sales or price earnings.
- **Macro economics indicators** such as economic surprises indices by region and globally.
- **Technical indicators** such as market breath indicator or put-call ratio.
- **Rates** such as 10 year us rate, 2 years yields or breakeven inflation information.

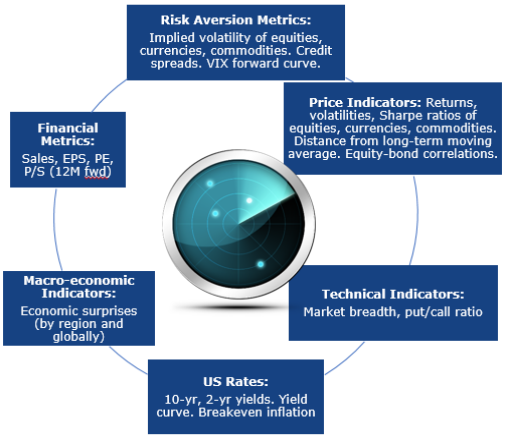

Figure 1. Probabilities of crash

2.3. Process of features selection

Using all the raw features would add too much noise to our model and would lead to bias decision. We thus need to select or extract the main meaning full features. As we can see in figure 2, we do so by removing the features in 2 steps.

- Based on gradient boosting trees, we rank the features by importance or contribution.
- We then pay attention to the severity of multicollinearity in an ordinary least squares regression analysis by computing the variance inflation factor (VIF) to remove co-linear features. Considering a linear model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + .. + \beta_n X_n + \epsilon$, the VIF is equal to $\frac{1}{1-R_j^2}$, with R_j^2 the multiple 2 for the regression of X_j. The VIF reflects all other factors that influence the uncertainty in the coefficient estimates.

At the end of this 2-part process, we only keep 33% of the initial dataset.

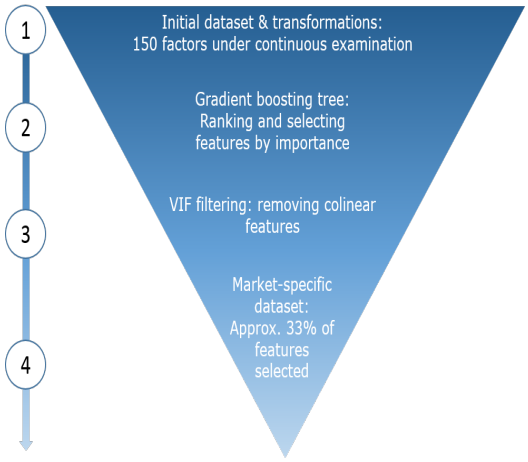

Figure 2. Probabilities of crash

It is interesting to validate that removing many data makes the model more robust and less prone to overfiting. In the next section, we will validate this point experimentally.

3. Results

3.1. Model presentation

Although our work is mostly describing the GBDT model, we compare it against common machine learning models. Hence we compare our model with four other models:

- RBF SVM that is a support vector model with a radial basis function kernel denoted and with a γ parameter of 2 and a C parameter of 1. We use the sklearn implementation. The two hyper parameters γ and C are found on the validation set.
- a Random Forest model whose max depth is taken to 1 and its boosted round to 500. On purpose, we take similar parameters as for our GBDT model so that we benefit from the averaging principle of taking a large boosted round and small decision trees. We found that for annual validation data set ranging from year 2015 on-wards and for the S&P 500 markets, the combination of a small max depth and a large number of boosted rounds performs well.
- a first deep learning model, referred in our experiment as Deep FC (for fully connected layers) that is naive built with three fully connected layers (64, 32 and one for the final layer) with a drop out in of 5 % between and Relu activation, whose implementation details rely on tensorflow keras 2.0
- a second more advance deep learning model consisting of two layers referred in our experiment as Deep LSTM: a 64 nodes LSTM layer followed by a 5% dropout followed by a 32 nodes dense layer followed by a dense layer with a single node and a sigmoid activation.

For both deep learning models, we use a standard Adam optimizer whose benefit is combine adaptive gradient descent with root mean square propagation (Kingma & Ba, 2014).

For each model, we train them either using the full data set of features or only the remaining features that are resulting from the features selection process as described in 2. Hence, for each model, we add a suffix ' raw' or ' FS' to specify if the model is trained on the full data set or after features selections. We provide the performance of these models according to different metrics, namely accuracy, precision, recall, f1-score, average precision, auc and auc-pr in table 1. The GBDT with features selection is among all metrics superior and outperform the deep learning model based on LSTM validating our assumption that on small and imbalanced data set, GBDT outperform deep learning models. In table 2, we compare the model with and without feature selection. We can see that using a lower and more sparse number of feature improves the performance of the model for the AUC and AUC pr metric.

3.2. AUC graphics

Figure 3 provides the ROC Curve for the two best performing models, namely the GBDT and the Deep learning LSTM model with features selection. Simply said, ROC curves enables to visualize and analyse the relationship between precision and recall and to stress test the model whether it makes more error of type I or error of type II when trying to

find the right answer. The receiver operating characteristic (ROC) curve plots the true positive rate (sensitivity) on the vertical axis against the false positive rate (1 - specificity, fall-out) on the horizontal axis for all possible threshold values. We can notice that the two curves are well above the *blind guess* benchmark that is represented by the dotted red line. This effectively demonstrates that these two models have some predictability power, although being far from a perfect score that will be represented by a half square. The ROC curve also gives some intuition whether a model is rather concentrating on accuracy or recall precision. In an ideal world, if the ROC curve of the model was above all other models' ROC curve, it will Pareto dominates all other and will be the best choice without any doubt. Here, we see that the area under the curve for the GBDT with features selection is 0.83 to be compared with 0.74 which is the one of the second best model, namely the Deep LSTM model with also Features selection. The curve of the first best model GBDT represented in blue is mostly over the one of the second best model the Deep LSTM model. This indicates that in most situations, we expect this model to perform better than the Deep LSTM model.

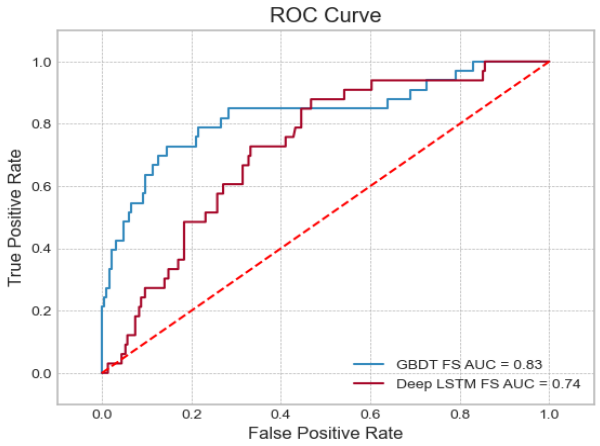

Figure 3. ROC Curve of the two best models

3.3. Dealing with imbalanced data

Machine learning algorithms work best when samples number in each class are about equal. However, when one or more classes are very rare, many models don't work too well at identifying the minority classes. In our case, we have very imbalanced class as the crisis regime only occurs 5 percents of the time. Hence the ratio between the normal regime and the crisis regime occurrence is 20! This is a highly imbalanced supervised learning binary classification and can not be done using standard accuracy metric. To avoid this drawback, first, we use the ROC AUC as a loss metric. The ROC AUC metrics is a good balance between precision and recall and hence accounts well for imbalanced data sets. We also weight more the crisis regime occurrence by playing with the *scale_pos_weight* parameter in LightGBM and set it to 20 which is the ratio between the class labeled 0 and the class labeled 1.

3.4. Out of sample probabilities

We provide in figure 4 the out of sample probabilities in connection with the evolution of the price of the S&P 500 merged back adjusted rolled future. In order to smooth the probability, we compute its mean over a rolling window of one week. We see that the probability spikes in end of February indicating a regime of crisis that is progressively turn down to normal regime in mid to end of March. Again in June, we see a spike in our crisis probability indicating a deterioration of market conditions.

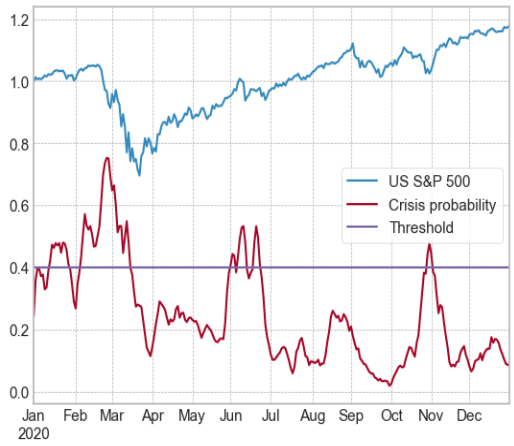

Figure 4. Mean over a rolling window of 5 observation of the probabilities of crash

3.5. Can it act as an early indicator of future crisis?

Although the subject of this paper is to examine if a crisis model is effective or not, we can do a simple test to check if the model can be an early indicator of future crisis. Hence, we perform a simple strategy consisting in deleveraging as soon as we reach a level of 40 % for the crisis probability. The objective here is by no means to provide an investment strategy as this is beyond the scope of this work and would require some other machine learning techniques likes the ones around deep reinforcement learning to use this early indicator signal as presented in (Benhamou et al., 2020f), (Benhamou et al., 2020b), (Benhamou et al., 2020e), (Benhamou et al., 2020c) or (Benhamou et al., 2020d).

The goal of this simple strategy that deleverages as soon as we reach the 40% threshold for the crisis probability is to validate that this crisis probability is an early indicator of future crisis. To be very realistic, we apply a 5 bps transaction cost in this strategy. We see that this simple

method provides a powerful way to identify crisis and to deleverage accordingly as shown by the figure 5. The logic of applying GBDT to filter out when to be long and when on the contrary to avoid a long position in the stock markets is similar in spirit to the work of filtering good and bad trades using XGBoost method as presented in (Saltiel & Benhamou, 2018a) with the general method presented in (Saltiel & Benhamou, 2018b).

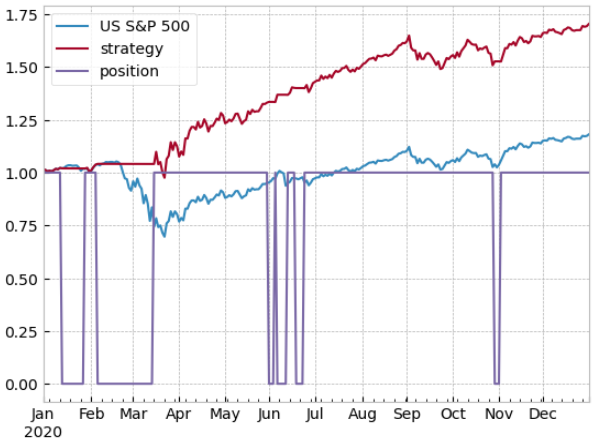

Figure 5. Simple strategy

4. Understanding the model

4.1. Shapley values

Understanding why the model makes a certain prediction can be as crucial as its prediction's accuracy. Using the work of (Lundberg & Lee, 2017), we use Shapley value to provide a fine understanding of the model. The Shapley value (Shapley, 1953) of a classifier is the average marginal contribution of the model over the possible different permutations in which the ensemble can be formed (Chalkiadakis et al., 2011). It is rigorously defined as follows.

Definition 4.1. ***Shapley value.*** *The Shapley value of binary classifier M in the ensemble $\mathcal{M}$, for the data point level ensemble game $G = (\mathcal{M}, v)$ is defined as*

$$\Phi_M(v) = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{M\}} \frac{|\mathcal{S}|!\,(|\mathcal{M}| - |\mathcal{S}| - 1)!}{|\mathcal{M}|!} (v(\mathcal{S} \cup \{M\}) - v(\mathcal{S})).$$

where $\mathcal{S}$ is the subset of features used, $v(\mathcal{S})$ is the prediction for features values in set $\mathcal{S}$ that are marginalized over features that are not included in set $\mathcal{S}$:

$$val(\mathcal{S}) = \int \hat{f}(x_1, ..., x_p) d\mathbb{P}_{x \notin \mathcal{X}} - \mathbb{E}[\hat{f}(X)]$$

Theoretically, calculating the exact Shapley value for a model is a hard task and should take $\mathcal{O}(|\mathcal{M}|!)$ time which is computationally unfeasible in large scale settings. However, a practical solution is to calculate the approximate Shapley values using the conditional expectation instead of the marginal expectation, which reduces computing time to $\mathcal{O}(|\mathcal{M}|^2)$ and is implemented in most gradient boosting decision trees library such as XGBoost or LightGBM or the Shap library: https://github.com/slundberg/shap

4.2. Shapley interpretation

We can rank the Shapley values by order of magnitude importance, defined as the average absolute value of the Shapley value over the training set of the model. Furthermore, the Shapley values are correlated with the feature, therefore enabling to highlight the nature and the strength of the relationship between the logit contribution and the feature. As a matter of fact, a positive correlation (resp. negative correlation) conveys a positive relationship between the feature evolution and the logit contribution. We provide here two figures to provide intuition about the model. Figure 6 displays the shap value sorted by order with the corresponding correlation and a color code to quantify if a feature is mainly increasing or decreasing the crisis probability. Figure 7 provides the full distribution and is commented in section 4.3.

Concerning figure 6, we find that the the most significant feature is the 250 days change in S&P 500 Price Earnings ratio (the forward 1 Yr Earnings of the index as provided by Bloomberg). Its correlation with the logit contribution is negative.

This relationship infers that a positive (resp. negative) change in P/E over one year lowers (resp. increases) the probability of crash. It indicates that a positive change in P/E over a one year horizon translates into a perception of improvement in the economic cycle and a positive regime for equities. This makes a lot of sense as it is well known that there are cycles of repricing whenever market participants anticipate a positive and rising regime in the equity markets. In short, when a majority of investors are optimistic, a higher valuation multiple prevail in equities' market

By the same token, a positive (resp. negative) change in the US 2 Yrs yield over 250 days characterizes a regime of growth (resp. recession) in equities. A positive change in the Bloomberg Base Metals index portrays a positive regime in equities, thus diminishing the probability of crash. The same reasoning applies for Fx Emerging Basket, S&P Sales evolution, the Euro Stoxx distance to its 200 moving average. Similarly, the EU Economic Surprise Index is used to characterize the economic cycle

Interestingly, the Machine Learning approach has identified the Put/Call ratio as a powerful contrarian indicator, using a low level of Put/Call ratio to detect a higher level of crash

in the S&P 500.

Indeed, a persistently low level of Put/Call ratio (as reflected by a low 20 days moving average) indicates an excessive bias of positive investors vs. negative investors and therefore an under-hedged market.

The correlation between the Put/Call ratio and the logit contribution is therefore negative, a low level of the Put/Call ratio involving an increase in the probability of crash. This makes a lot of sense. A higher (resp. lower) Risk Aversion accounts for a higher (resp. lower) crash probability and the same relationship is verified with the realized 10 days S&P 500 volatility.

Last but not least, the Nasdaq 100 is used as a contrarian indicator : the higher the percentage of Nasdaq and the Sharpe Ratio, the higher the crash probability. In general, the trend of other markets is used in a pro cyclical ways (Euro Stoxx, BCOM Industrials, FX emerging) whereas the domestic price indicators are used in a contrarian way (Nasdaq 100, S&P 500). This is where we can see some strong added value of the machine learning approach that mixes contrarian and trend following approaches while human favor mostly one single approach.

4.3. Shapley Values' Distribution

Because some of the features have a strong non linear behavior, we also provide in figure 7 the full marginal distribution. More precisely, figure 7 displays a more precise relationship between the Shapley values and the whole distribution of any individual features.

For instance, high 250 days change in P/E ratio represented in red color has a negative impact on the logit contribution, everything else being equal. Therefore, an increase in the P/E ratio involves a decrease in the crash probability of S&P 500 and vice versa. The dependency of the crash probability is similar for the change in US 10 Yrs and 2 Yrs yield: the higher (resp. lower) the change in yield, the lower (resp. higher) the crash probability of S&P 500.

However, the dependency on BCOM Industrial Sharpe ratio calculated over 120 days is more complex and non linear. As a matter of fact, low Sharpe ratio of industrial metals can have conflicting effects on the crash probability either increasing or decreasing the probability whereas elevated Sharpe ratio has always a negative impact on the crash probability. The same ambiguous dependency is observed against the Sharpe ratio of FX Emerging calculated on a 100 days horizon. This behavior confirms the muted correlation between the FX EM Sharpe ratio and the Shapley value although the variable is significant. This complex dependency highlights the non linear use of the feature by GBDT models and the interaction between this feature with other features uses by the model. By the same token, the change in Sales of S&P 500 over 20 days has not a straightforward relationship with the crash probability. First of all, mostly elevated values of the change in sales are used by the model, shedding light on the conditional use of extreme values of the features by the GBDT model. Furthermore, elevated changes in S&P 500 sales over 20 days are mostly associated with a diminution of the crash probability but not in every instance.

The use of the distance to the Euro Stoxx 50 to its 200 days moving average is mostly unambiguous. Most of elevated levels in the feature's distribution involves a decrease in the crash probability whereas weak levels conveys a bear market regime and therefore accounts for an increase in the crash probability. Meanwhile, some rare occurrences of elevated values of the distance in Euro Stoxx prices' to their 200 days moving average can be associated with higher probability of crash highlighting a non linear dependency. The 20 days Moving Average of the Put/Call Ratio is used as a contrarian indicator: low values of the indicator reflects an under hedged market and convey an increase in the crash probability whereas elevated values carry a regime of extreme stress where hedging strategies prevail thus accounting for a decrease in the crash probability. This finding is consistent with the correlation of -0.88 between the Put/Call ratio and the Shapley Values as showed in figure 6. The relationship between the 20 days moving average of Risk Aversion and the logit contribution is also clearly negative: above all, lower values of Risk Aversion are related to negative contribution to crash probability, whereas higher Risk Aversion accounts for an increase in the crash probability. This relationship is consistent with the correlation of -0.89 displayed in figure 6.

The use of the change in Nasdaq 100 price over 20 days is confirmed as a contrarian indicator (correlation of -0.89 in figure 6). As illustrated in Figure 7, negative returns of the Nasdaq 100 over 20 days are associated with lower crash probability, everything else being equal. Conversely, the most elevated values of Nasdaq 100 20 days returns produces an increase in the crash probability but in a more muted way. Figure 7 therefore provides an additional information: negative returns of the Nasdaq 100 are more used than positive returns in the forecast of crash probability. Conversely, the 20 days Euro Stoxx returns is used in a pro cyclical way as inferred by the correlation -0.85 displayed in figure 6. Higher (resp. lower) Euro Stoxx returns are associated with a decline (resp. surge) in the crash probability. As previously stated, the GBDT model uses non US markets in a procyclical way but US markets in a contrarian way and as displayed in figure 7, the type of relationship seems to be univocal.

4.4. Can the machine learning provide an answer to the Covid Equity bubble?

Not only can Shapley values provide a global interpretation as described in section 4.2 and 7, it can also supply a local interpretation at every single date. Hence we have 13 figures ranging from 8 to 20. These figures provide the monthly evolution of the Shapley value over 2020. We can notice that a lot of features are the same from months to months, indicating a persistence of behavior and importance of features like SP 500 Price Earning percentage over 120 days, risk aversion, economical cycles variables like industrial metals and other equity markets as well as central bank influenced variables like nominal and real rates and some technical indicator like put call ratio.

On 1st January 2020, the model was still positive on the S&P 500 as the crash probability was fairly low, standing at 9.4%. The positive change in P/E at 6% accounted for a decrease in the probability, while a risk aversion reflected ample liquidity and positive EU Economic Surprise index all reinforced a low probability crash. However, the decline in the US LIBOR is characteristic of a falling economy, thus increasing the crash probability. Similarly, the elevated Put/Call ratio reflected excessive speculative behavior. At the beginning of February, the probability, though still moderate, started to increase slightly. Yet, at the onset of the Covid crash, probability increased dramatically on the back of deteriorating dynamics of industrial metals, falling euro stoxx prices, declining FTSE prices, degradation of EU economic surprises and failing S&P 500 P/E. In a nutshell, the model identified a downturn in the equities' cycle. This anticipation eventually proved prescient. Meanwhile, at the start of April 2020, the model eased the crash probability. The Nasdaq Sharpe ratio appeared excessively negative, the Put/Call ratio displayed extremely prudent behavior among investors. Contrarian indicators eventually started to balance pro cyclical indicators, therefore explaining the easing of the crash probability. During several months, the crash probability stabilized between 20% and 30% until the start of July which showed a noticeable decline of probability towards 11.2%. The P/E cycle started to improve and negative signals on base metals and other equities' dynamics started to improve to the upside. Although the crash probability fluctuated, it remained contained though out the rest of 2020.

At the turn of the year 2020, most of signals were positive on the back on improving Sharpe ratio of Industrial metals, failing dollar index, easing of Risk Aversion, reflecting ample liquidity in financial markets, convalescent other equities markets. For sure, this improving backdrop is moderated by lower rates over one year and various small contributors. Meanwhile, the features' vote leans towards the bullish side.

This rationalization of the post equity bubble does not provide an excuse of the absolute level of equity prices. Nonetheless, the dynamics of equity prices can clearly be explained in light of past crises and improving sentiment. For sure, sentiment may have been driven by unprecedented fiscal and monetary interventions but the impact they had on markets could have been successfully analyzed by a machine learning approach learning only from pretended episode. Therefore, equity prices may be irrational at the turn of 2020 but dynamics of prices were nonetheless rational from a machine learning perspective.

In summary, machine learning does provide an answer thanks to a detailed analysis of the different features. It does spot that given the level of various indicators and in particular industrial metals, long terms yield, break even inflation, that reflect public intervention and accommodative monetary policies of central banks that mute and ignore any offsetting factors like lower rates over one year, the model forecast a rather low probability of a large correction over 15 days.

However, one must be careful and should not be overconfident about the model forecast. The model presented in the paper has a short time horizon (15 days), which does not portend any equity evolution on a longer time frame. It may miss certain behavior or new relationships between markets as it only monitors 150 variables. More importantly, the model reasons using only past observations. Should the future be very different from the past, it may wrongly compute crash probabilities influenced by non repeating experience.

5. Conclusion

In conclusion, in this work, we see that GBDT methods can provide a machine learning answer to the Covid Equity Bubble. Using a simple approach of two modes, GBDT is able to learn from past data and classify financial markets in normal and crisis regimes. When applied to the S&P 500, the method gives high AUC score providing some evidence that the machine is able to learn from previous crisis. We also report that GBDT report improved accuracy over other ML methods, as the problem is a higly imbalance classification problem with a limited number of observation. The analysis of Shapley values caters valid and interesting explanations of the current Covid equity high valuation. In particular, the machine is able to find non linear relationships between different variables and detects the intervention of central banks and their accommodative monetary policy that somehow inflated the current Covid Equity bubble.

References

Aguilar-Rivera, R., Valenzuela-Rendón, M., and Rodríguez-Ortiz, J. Genetic algorithms and darwinian approaches in financial applications: A survey. *Expert Systems with*

Applications, 42(21):7684–7697, 2015. ISSN 0957-4174.

Atsalakis, G. S. and Valavanis, K. P. Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36(3, Part 2):5932–5941, 2009.

Ballings, M., den Poel, D. V., Hespeels, N., and Gryp, R. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20): 7046–7056, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.05.013.

Benhamou, E. Trend without hiccups: a kalman filter approach. *IFTA Journal*, 2018:38–46, 2018.

Benhamou, E., Atif, J., and Laraki, R. A discrete version of cma-es. *ArXiv*, cs.LG:1812.11859, 2019a.

Benhamou, E., Saltiel, D., Guez, B., and Paris, N. Bcma-es ii: revisiting bayesian cma-es. *ArXiv*, cs.LG:1904.01466, 2019b.

Benhamou, E., Saltiel, D., Verel, S., and Teytaud, F. Bcma-es: A bayesian approach to cma-es. *ArXiv*, cs.LG: 1904.01401, 2019c.

Benhamou, E., Saltiel, D., Laraki, R., and Atif, J. BCMA-ES: a conjugate prior Bayesian optimization view. working paper or preprint, October 2020a. URL `https://hal.archives-ouvertes.fr/hal-02977523`.

Benhamou, E., Saltiel, D., Ohana, J., Atif, J., and Laraki, R. Deep reinforcement learning (DRL) for portfolio allocation. In Dong, Y., Ifrim, G., Mladenic, D., Saunders, C., and Hoecke, S. V. (eds.), *ECML PKDD 2020*, volume 12461 of *Lecture Notes in Computer Science*, pp. 527–531. Springer, 2020b.

Benhamou, E., Saltiel, D., Ohana, J.-J., and Atif, J. Detecting and adapting to crisis pattern with context based deep reinforcement learning, 2020c.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Time your hedge with deep reinforcement learning, 2020d.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Aamdrl: Augmented asset management with deep reinforcement learning. *arXiv*, 2020e.

Benhamou, E., Saltiel, D., Ungari, S., and Mukhopadhyay, A. Bridging the gap between markowitz planning and deep reinforcement learning, 2020f.

Brown, I. and Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012. ISSN 0957-4174.

Chalkiadakis, G., Elkind, E., and Wooldridge, M. Computational Aspects of Cooperative Game Theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 3146–3154. Curran Associates, Inc., 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization, 2014.

Krauss, C., Do, X. A., and Huck, N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702, 2017.

Li, Y. and Ma, W. Applications of artificial neural networks in financial economics: A survey. In *2010 International Symposium on Computational Intelligence and Design*, volume 1, pp. 211–214, 2010.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions, 2017.

Marceau, L., Qiu, L., Vandewiele, N., and Charton, E. A comparison of deep learning performances with others machine learning algorithms on credit scoring unbalanced data. *CoRR*, abs/1907.12363, 2019.

Mehra, R. and Prescott, E. The equity premium: A puzzle. *Journal of Monetary Economics*, 15(2):145–161, 1985.

Nti, I. K., Adekoya, A. F., and Weyori, B. A. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, pp. 1–51, 2019.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 6638–6648. Curran Associates, Inc., 2018.

Rundo, F., Trenta, F., di Stallo, A. L., and Battiato, S. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.

Saltiel, D. and Benhamou, E. Trade Selection with Supervised Learning and OCA. *arXiv e-prints*, art. arXiv:1812.04486, December 2018a.

Saltiel, D. and Benhamou, E. Feature selection with optimal coordinate ascent (OCA). *arXiv e-prints*, art. arXiv:1811.12064, November 2018b.

Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. *arXiv preprint arXiv:1911.13288*, 2019.

Shah, D., Isah, H., and Zulkernine, F. Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2):26, 2019.

Shapley, L. S. A Value for n-Person Games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Xing, F. Z., Cambria, E., and Welsch, R. E. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.

A. Models comparison

Table 1. Model comparison

Model	accuracy	precision	recall	f1-score	avg precision	auc	auc-pr
GBDT FS	**0.89**	**0.55**	**0.55**	**0.55**	**0.35**	**0.83**	**0.58**
Deep LSTM FS	0.87	0.06	0.02	0.05	0.13	0.74	0.56
RBF SVM FS	0.87	0.03	0.07	0.06	0.13	0.50	0.56
Random Forest FS	0.87	0.03	0.07	0.04	0.13	0.54	0.56
Deep FC FS	0.87	0.01	0.02	0.04	0.13	0.50	0.56
Deep LSTM Raw	0.84	0.37	0.33	0.35	0.21	0.63	0.39
RBF SVM Raw	0.87	0.02	0.01	0.05	0.13	0.50	0.36
Random Forest Raw	0.86	0.30	0.09	0.14	0.14	0.53	0.25
GBDT Raw	0.86	0.20	0.03	0.05	0.13	0.51	0.18
Deep FC Raw	0.85	0.07	0.05	0.02	0.13	0.49	0.06

Table 2. Difference between model with features selection and raw model

Model	accuracy	precision	recall	f1-score	avg precision	auc	auc-pr
GBDT	0.02	0.35	0.52	0.49	0.23	0.32	0.41
Deep LSTM	0.03	- 0.31	- 0.31	- 0.30	- 0.08	0.11	0.17
RBF SVM	-	0.01	0.06	0.01	-	-	0.20
Random Forest	0.02	- 0.27	- 0.02	- 0.10	- 0.02	0.01	0.31
Deep FC	0.02	- 0.06	- 0.03	0.02	-	0.01	0.50

B. Models Understanding with Shapley values

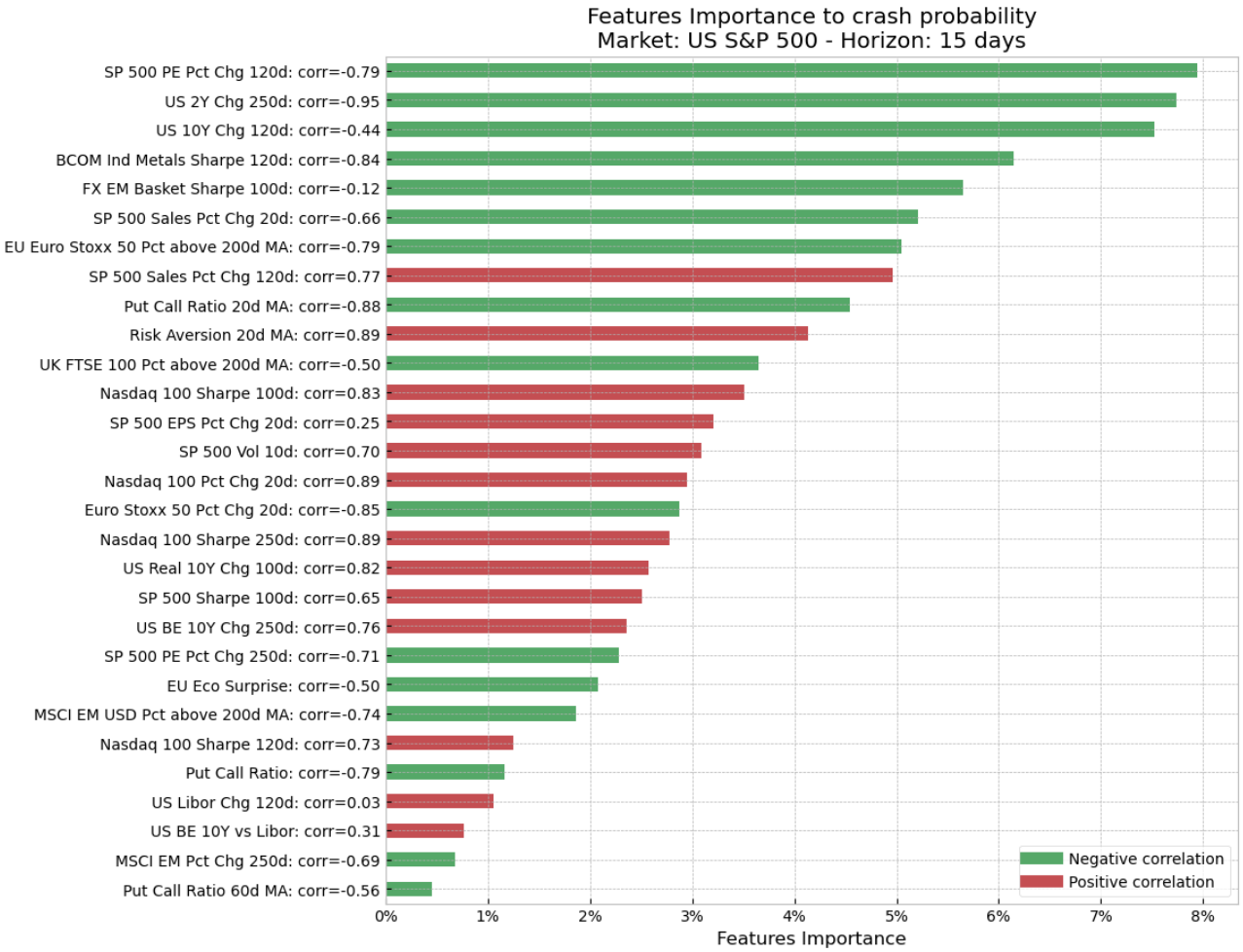

Figure 6. Marginal contribution of features

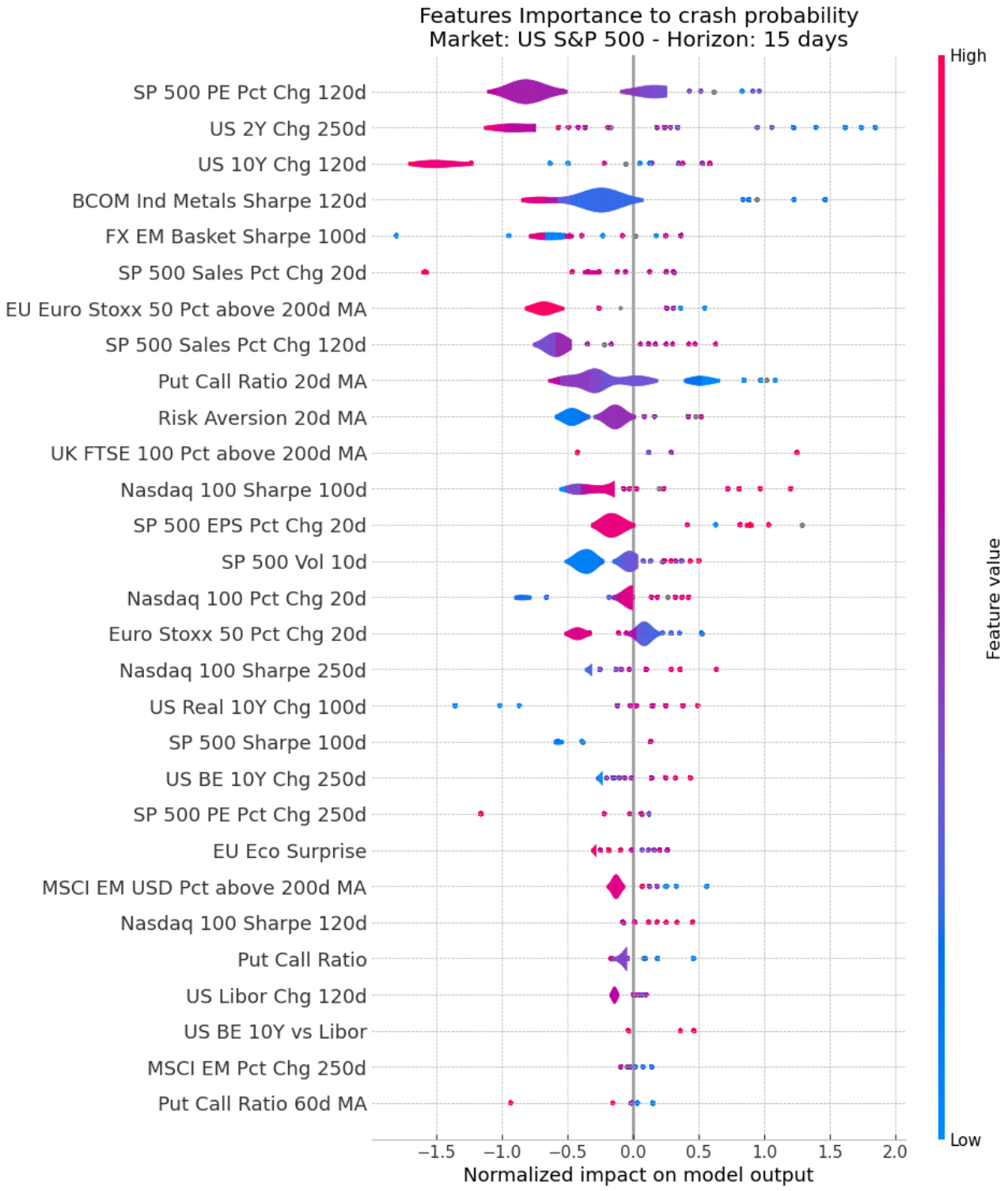

Figure 7. Marginal contribution of features with full distribution

Figure 8. Shapley values for 2020-01-01

Figure 9. Shapley values for 2020-02-03

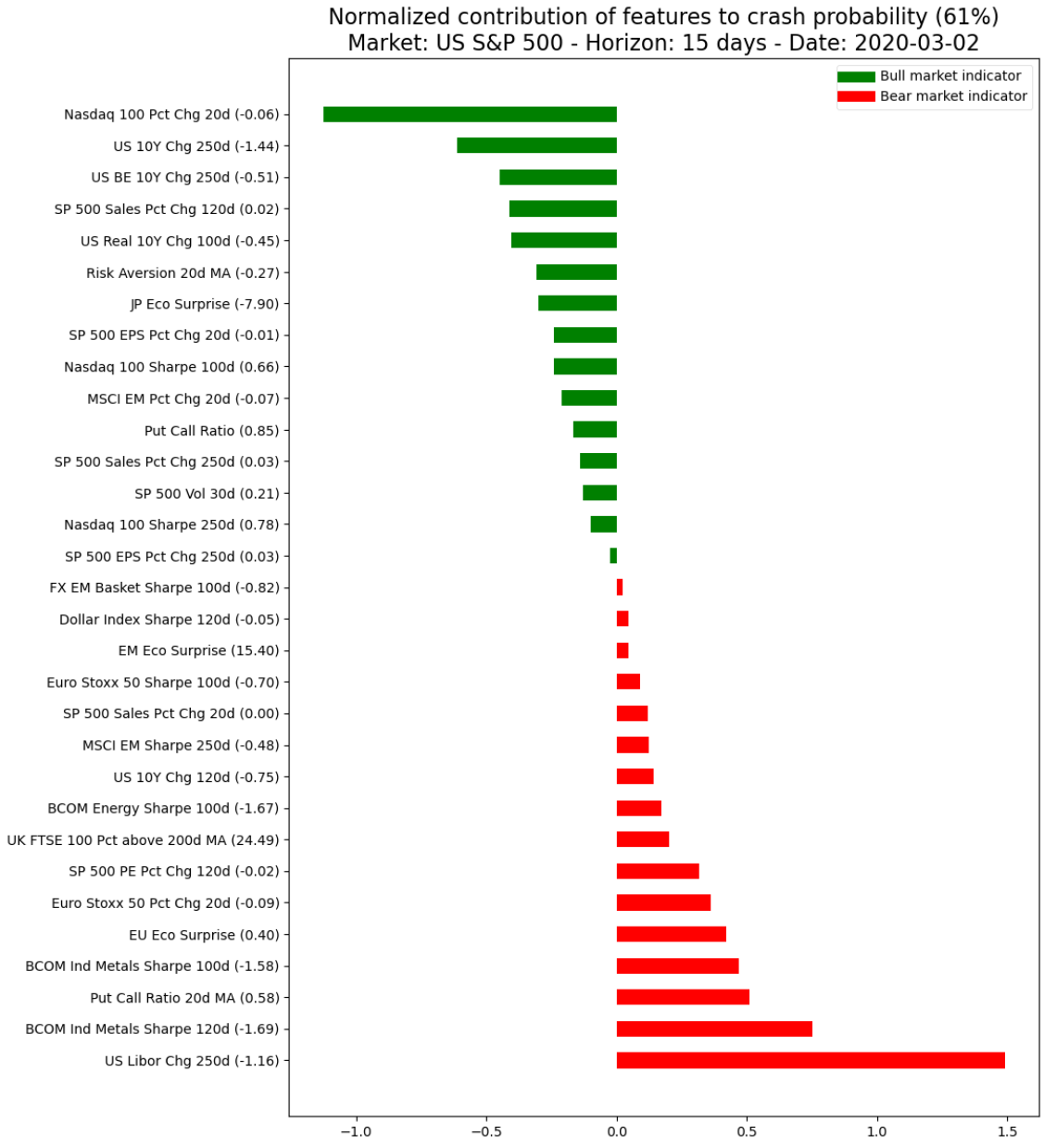

Figure 10. Shapley values for 2020-03-02

Figure 11. Shapley values for 2020-04-01

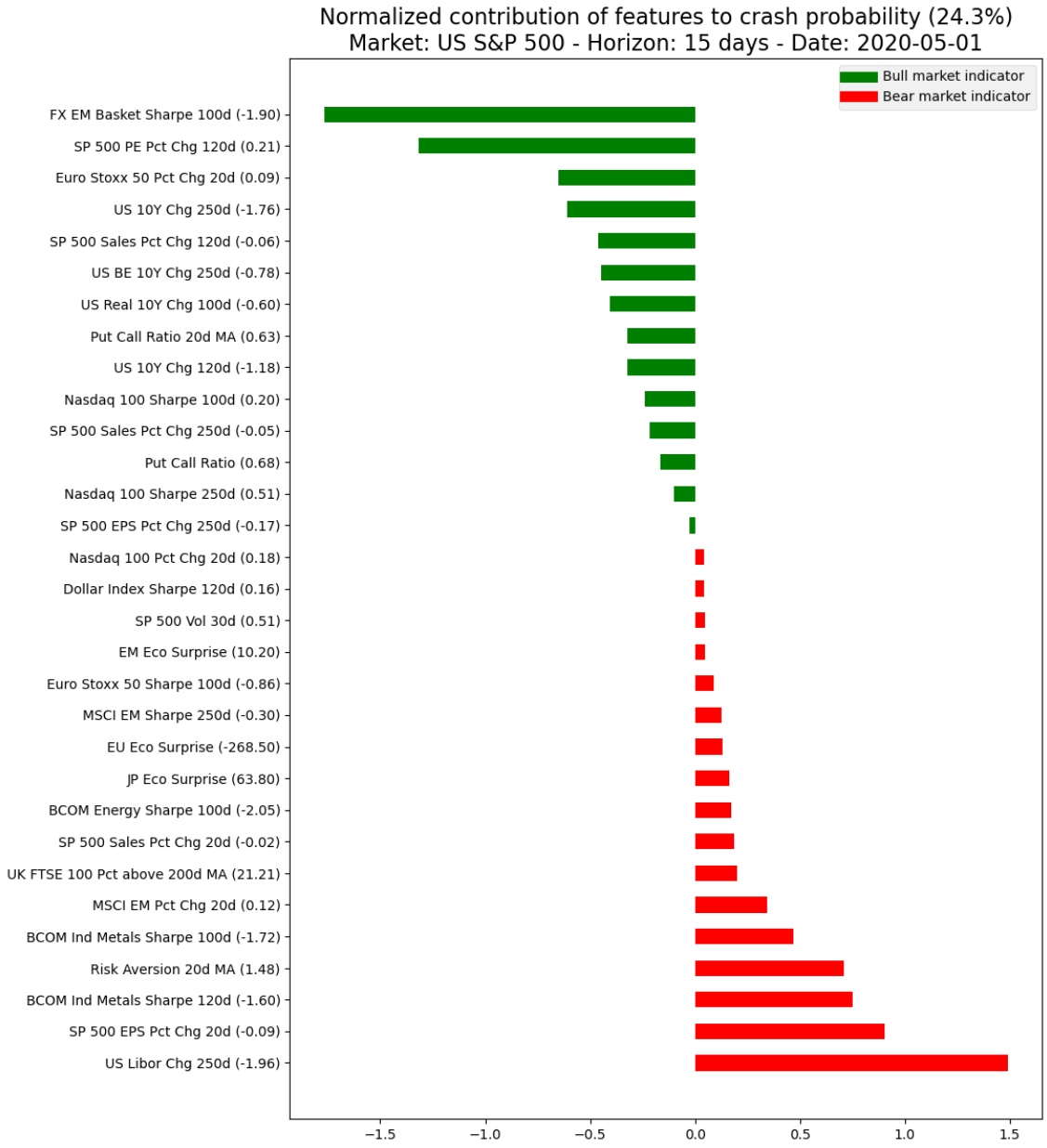

Figure 12. Shapley values for 2020-05-01

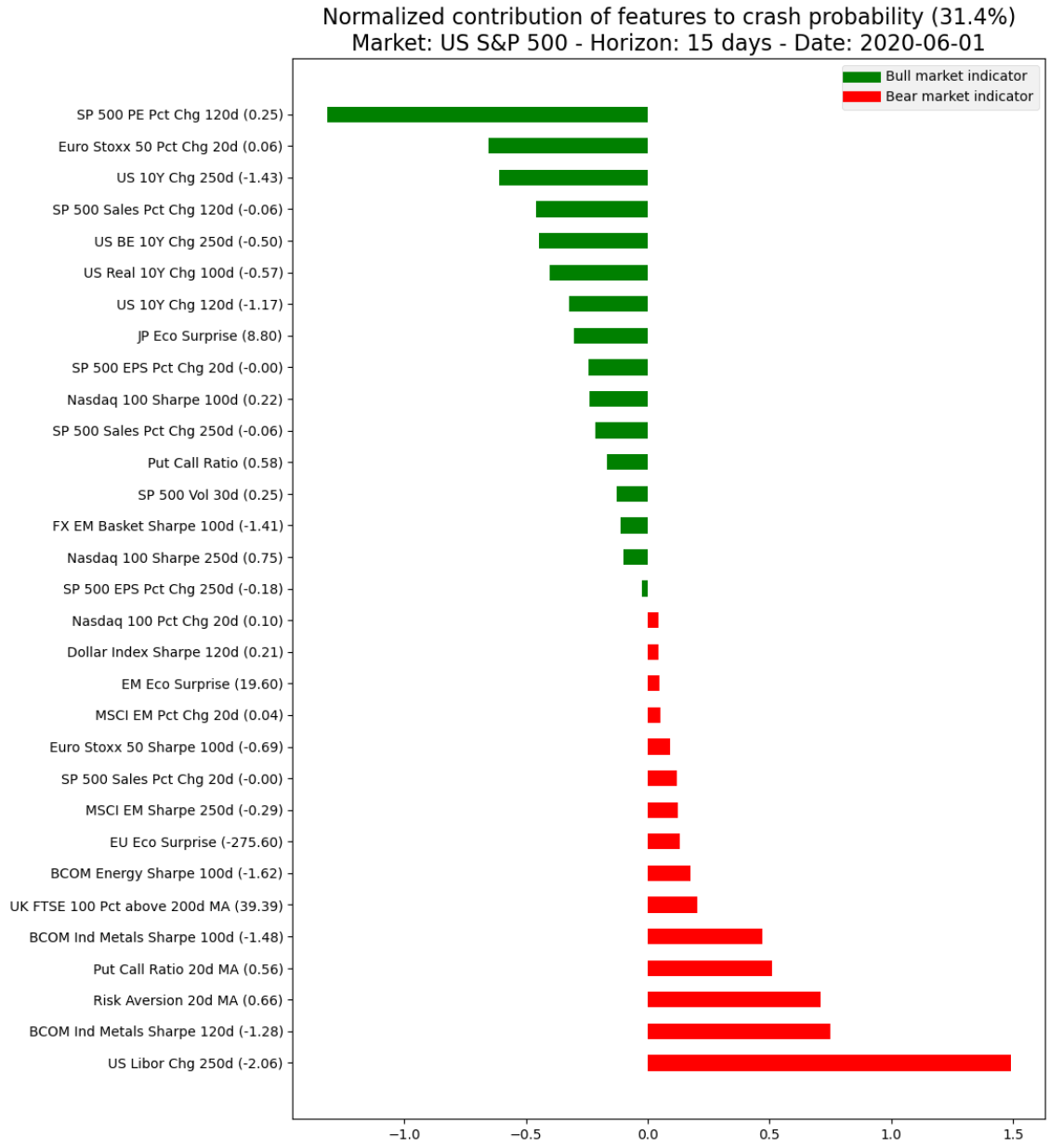

Figure 13. Shapley values for 2020-06-01

Figure 14. Shapley values for 2020-07-01

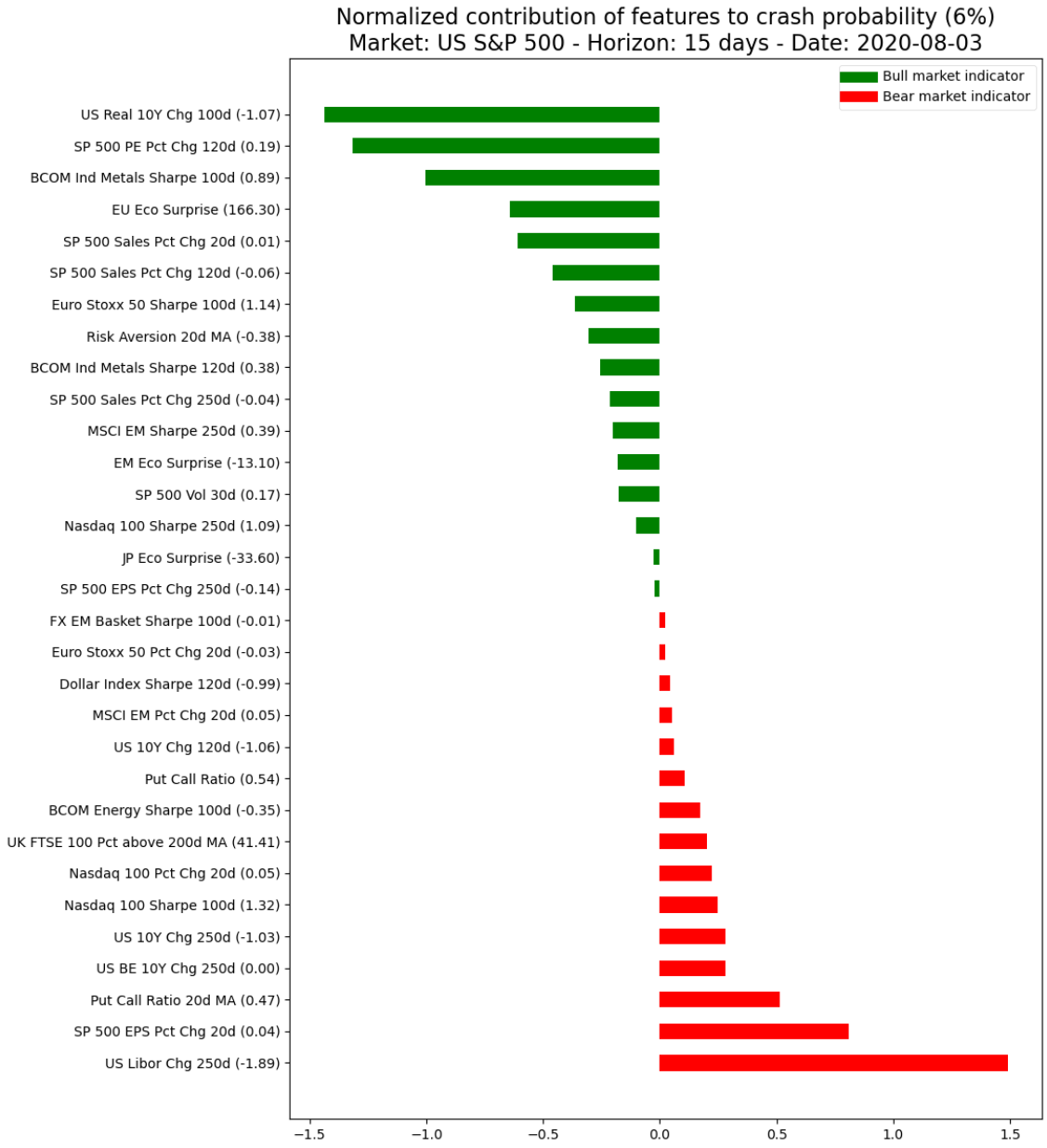

Figure 15. Shapley values for 2020-08-03

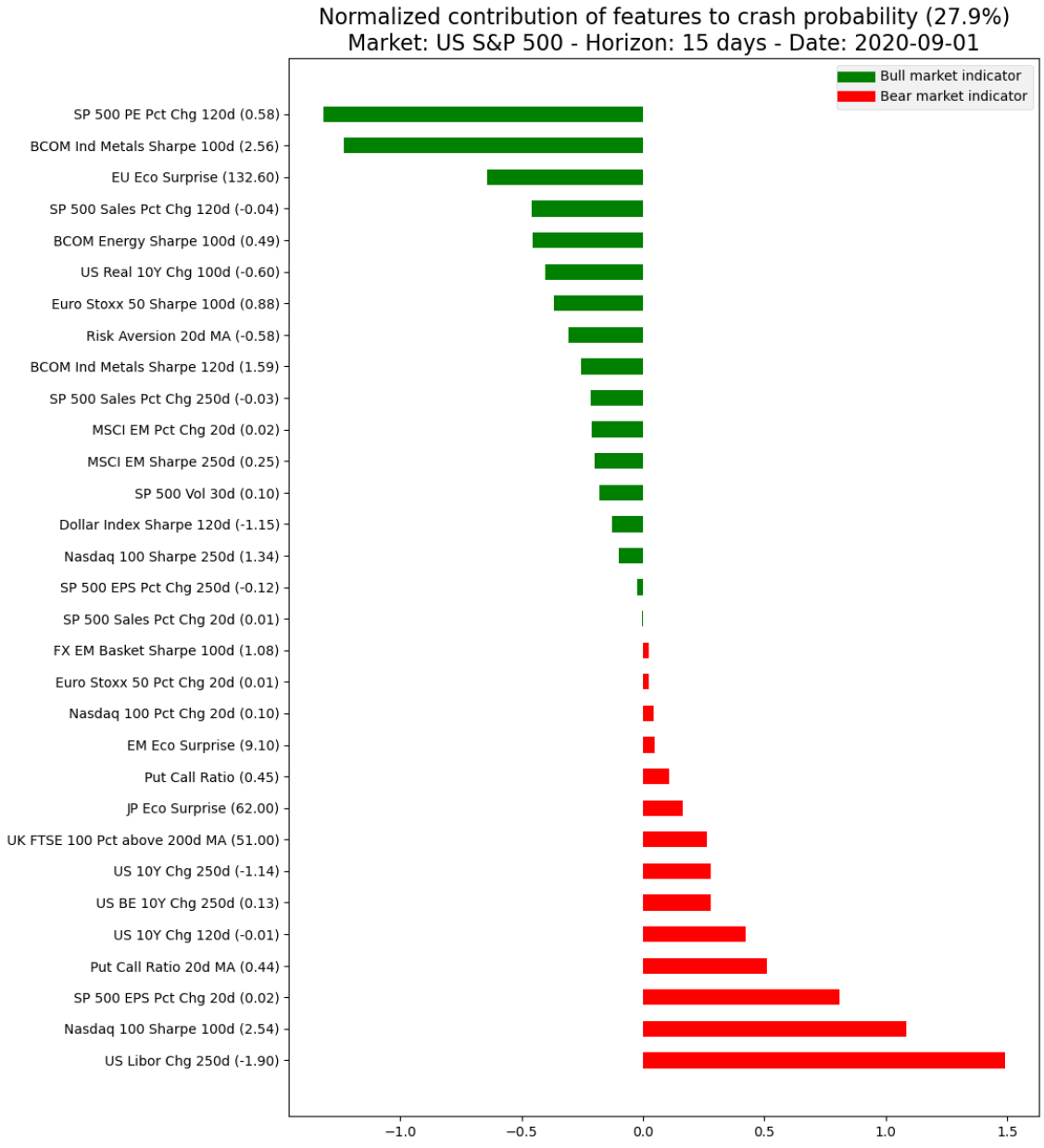

Figure 16. Shapley values for 2020-09-01

Figure 17. Shapley values for 2020-10-01

Figure 18. Shapley values for 2020-11-02

Figure 19. Shapley values for 2020-12-01

Figure 20. Shapley values for 2020-12-31