Sentiment-Based Trading Algorithm: A Reddit Case

Sebastien Gorgoni
Faculty of Business & Economics
University of Lausanne
Lausanne, Switzerland
sebastien.gorgoni@unil.ch

Liam Svoboda
Faculty of Business & Economics
University of Lausanne
Lausanne, Switzerland
liam.svoboda@unil.ch

***Abstract*—With the latest market turmoils involving the subreddit r/wallstreetbets, regarding GameStop Corporation and AMC Entertainment, the American social news aggregation Reddit, or the *front page of the internet* as it claims to be, has been a place of choice for young retail investors to find new trendy investment propositions. Simultaneously, as governments were enforcing stay-at-home policies amid the COVID pandemic, the number of retail investors, investing through their smartphone or laptop, were on the rise due to boredom. Therefore, this paper aims to create an event-based trading algorithm, by predicting the direction of an asset's price movement, using data collected from Reddit as well as historical financial time series.**

***Index Terms*—Reddit, sentiment-based trading algorithm, news-based trading, r/wallstreetbets, Gamestop, S&P500, Tesla, Bitcoin**

I. Introduction

Reddit is a network of forums that covers almost any subject imaginable in the form of videos and pictures to news and open discussions. Its users can post anything they want, as long as it is appropriate for the particular section. They can up-vote or down-vote posts as they see fit and also comment on them. The website contains many sections called ”subreddits”, which gather all posts regarding one particular topic and at which users can become members. There are all matter of subreddits, from more legitimate ones such as r/MachineLearning with 1.9m members to more questionable ones such as /r/monkslookingatbeer dedicated solely pictures of monks looking at beer which gathers a surprising 37k people. From what started as a small forum in 2005, Reddit became one of the most influential websites of the 21st century, within the ranks of major social networks such as Facebook or Twitter for example.

This ability to concentrate large numbers of users around a precise topic coupled with the democratization of financial markets which allow more individuals to access investment products led to the creation of an hive mind with the ability to move markets. This was only exacerbated with the advent the pandemic which led to people spending more time on the internet and having more money for discretionary spending. All these factors combined led to a perfect storm for an unprecedented event to occur on financial markets which would shake the world of finance and attract mass attention from the mainstream media. The event in question took place at the beginning of 2021 when users of the r/wallstreetbets (a subreddit where predominantly young and risk-seeking people discuss stocks, crypto-currencies and option trading) forced a short-squeeze on the of GameStop Corporation (GME) stock following the discovery of a disproportionately high short interest held by notorious hedge funds which deemed the stock undervalued. The consequences of this were staggering: a stock barely trading at 15$ in December of 2020 rocketed to 480$ in January of 2021. If online brokers made the stock market accessible to the average Joe, this event made him interested. What followed was a monumental spike in the number of members of the subreddit as displayed by figure 1 making it a force to be reckoned with and also a surge in interest for financial markets with now newly made retail investors eager to pounce on any occasion they get to participate in what they hope to be the next occurrence of such an event. Very recently, the same event happened to AMC Entertainment, in a similar way as with GME. With public figures, most notable of which is Elon Musk, the CEO of Tesla Inc. and SpaceX, becoming persons of reference and blind trust of the r/wallstreetbets community. Indeed, a single tweet or statement could trigger the price of Bitcoin, Dogecoin, Tesla and so on to fluctuate wildly.

Having witnessed these extraordinary events, and the involvement of the Reddit community, we wanted to see if it is possible to predict movements in certain assets by deriving the sentiment from a handful of key threads and coupling this information with some other more traditional technical indicators. We decided to focus on the S&P500 index, Tesla Corp. and Bitcoin, as the first is the gold-standard equity market index and the last two are greatly discussed on the website.

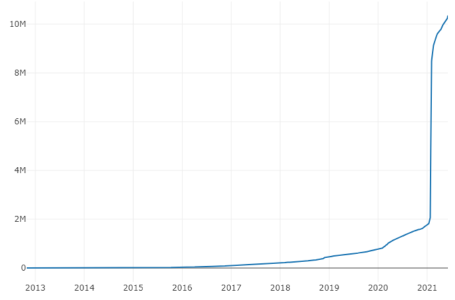

Fig. 1. Evolution of r/wallstreetbets members
Source: https://subredditstats.com/r/wallstreetbets

II. Research Question and Relevant Literature

The idea of predicting stock market movements through news is a topic which has been the focus of many research papers as it puts into question the validity of the Efficient Market Hypothesis (EMH), which suggests that prices reflect all public information [1] (Fama, 1970) and that information is reflected through price without delay (Malkiel, 2003) [2]. As shown by the primary research conducted by Oberlechner and Hocking (2004) [3], there is evidence that market participants incorporate the anticipated impact of new information in making decisions. In addition, Leinweber and Sisk (2011) [4] show that a trading strategy that uses news headlines analysis would generate positive alpha. Furthermore, Mehtab and Sen (2020) [5] conclude that sentiment analysis does give a significantly improved input to classical regression or classification approaches.

As the availability of information continues to surge, machine learning techniques such as deep learning for example have become increasingly applied to many different fields. Indeed, their ability to solve complex nonlinear problems make their applications appear limitless, ranging from self-driving cars to translation and astronomy to marketing and most importantly for this paper, stock price prediction (Akita, Yoshihara, Matsubara, 2016) [6]. Researchers have been able to leverage big data techniques, especially in the deep learning field, to predict market price movements (Tsai, Hsiao, 2010) [7]. One branch of machine learning we will be using is Natural Language Processing (NLP). Originating decades ago (Turing, 1950) [8] its goal is to undertake the not so obvious task of rendering language, which humans can manipulate and comprehend with ease, into a purely mathematical format to be digested by a computer for use in optimisation problems for example. One such method is called word embedding. Popularized by Mikolov et al. (2013) [9] with word2vec, this NLP technique project words in vector space and has the ability to capture intricate meaning and relation between words. Another, arguably less sophisticated approach which we will be taking consists in labeling words according to how they will be used. In the context of our project, we make use of various sentiment analysis techniques which come with built-in lexicons which is sentiment-labeled.

Much of the recent research has been dedicated to performing sentiment analysis to detect influence on stock market fluctuations using Twitter thanks to the swathes of available information retrievable through the Twitter API. It was found that simple word embedding techniques have shown to outperform other more advanced ones, for instance convolutional neural networks (Jermann 2017) [10]. However, most of the previous works using Twitter data have not performed significantly better than flipping a coin (Wai, Peng, 2018) [11]. One reason could be that using Twitter data could present a lot of noise and is rarely focused on one subject. Therefore, this research focuses on using Reddit posts across various judicious subreddits and determine their sentiments.

We have decided to use our models on the S&P500 index, Tesla Corp. and Bitcoin as they are probably affected differently by macroeconomic factors and news and thus the same news headlines would have different prediction power for different asset classes. For example, Foreign Exchange and equity market have been regarded as liquid and very sensitive to macro market sentiment (Hu, Zhao, Khushi, 2021) [12]. Taking different asset classes aims to understand if different sensitivity to macro environment of asset classes would change the predictivity. In addition, the two last assets are widely discussed on the subreddits we chose meaning that we expected to gather more relevant and specific information pertaining to them.

III. Methodology

A. Prediction Models

In order to maximise our chances of success, we trained 8 different types of supervised machine learning models with varying degrees of complexity. What follows is a brief description of each one:

K-Nearest Neighbours: One of the more primitive classification models, the k-Nearest Neighbours method looks at the k closest data points of the observation we aim to classify and assigns the point into the class most present amongst its neighbours. The underlying logic being that points which are closer together are expected to have the same characteristics.

Support Vector Machine: This method aims to find a hyperplane to use as a decision boundary for class separation. Although many such boundaries can potentially exist, here, the distance between the boundary and its nearest points is maximized to allow for unseen data points to be correctly classified with better precision.

Random Forest: The Random Forest classifier is a bagging ensemble classifier meaning that it combines independent simpler models for an increase in performance over the individual. The building block of this classifier is the Decision Tree which is described by a series of binary classifications branching out as to discriminate observations into classes.

Individually, the trees are prone to over-fitting but, when combined into a Random Forest the variance is reduced and therefore a better precision compared to a standalone Tree.

The Random Forest samples randomly from the training set, trains a specified number of Decision Trees and averages the predictions given by each tree to come to a final conclusion. The result of this process is a reduced variance and therefore a better precision compared to a standalone Tree.

Adaptive Boosting: Similar to the concept of a random forest in that it is also a meta-estimator often based on decision trees, it differs on the basis that Adaptive Boosting uses a boosting ensemble method. Indeed, it generates trees one by one and adapts them to become incrementally better than the previous one as opposed to bagging where multiple trees are built simultaneously and the results are averaged out like in the Random Forest.

Naive Bayes: One of the more primitive and therefore computationally cost effective models, Naive Bayes, applies

Bayes' theorem to calculate the conditional probability that a given point belongs to a class.

Logistic Regression: This regression-based classifier makes use of a sigmoid function, more specifically the namesake logistic function, which maps real numbers to a value between 0 and 1 to be used as an activation function in order to perform a binary classification. The model is trained to calibrate weights for the dependant variables which make the output of the logistic function determine the best class for the data point by comparing it to a threshold value (typically 0.5) which separates the two classes.

Deep Neural Network: Building upon a similar concept as the logistic regression, the Deep Neural Network (DNN) stacks and layers multiple activation functions so as to capture more subtleties in the classification problem.

In our case, the inputs are entered into the model, Rectified Linear Unit (ReLU) activation functions are used to activate the first hidden layer consisting of 32 neurons. These in turn can activate the second hidden layer of 16 neurons through ReLU functions of their own. Finally, the resulting 16 activations are fed through a sigmoidal activation function to the output layer representing the target variable. We also added dropout layers which ignore a certain proportion of the data between layers to mitigate over-fitting. An adam optimization method and a binary cross-entropy loss function were selected for our DNN.

Long Short-Term Memory Network: Finally, the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) [13] Network is by far the most intricate and in theory the best suited model which makes it deserving of a more in-depth explanation.

The reason for its usefulness in our application is its ability to capture relations between sequential observations, an important characteristic in the context time-series analysis of financial assets which often display trends.

LSTM is a type of Recurrent Neural Network (RNN). These have the ability to pass on information from the hidden layer resulting from the passing of one data point in one time-step to the hidden layer of the next time-step to be used in conjunction with the input variables. RNNs in their simpler forms can suffer from vanishing or exploding gradients which cause information to either be lost or become amplified throughout long-term dependencies, ultimately leading to erroneous predictions.

LSTMs solve this issue by essentially creating multiple compartments which are trained to let essential information through and useless information be forgotten. The difference between vanilla RNNs and LSTMs is illustrated in figure 2 and figure 3.

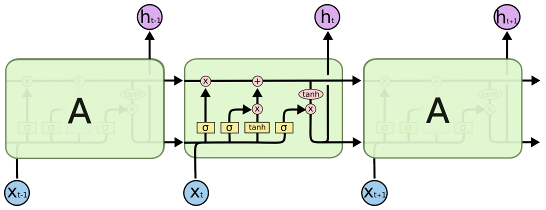

Fig. 2. Basic LSTM
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

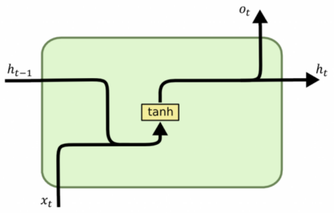

Fig. 3. Basic RNN
Source: https://www.baeldung.com/cs/nlp-encoder-decoder-models

We can distinguish 4 main elements in an LSTM module the cell state and 3 gates which control the flow of information.

1) The cell state C_t permeates the entire chain of time-steps potentially allowing long-term dependencies.
2) The forget gate f_t, a sigmoid function decides whether to let the C_{t-1} impact the contemporaneous cell C_t by taking as inputs h_{t-1} the remnant information from the previous time-step and x_t the data from the current time-step.
3) The input gate (i_t), a sigmoid activation function, along with a tanh activation function determine how to update C_{t-1} into C_t. The sigmoid dictates which part must be altered and the tanh tells us by how much. Both functions take h_{t-1} and x_t as inputs.
4) Finally, h_t is computed much like in the previous point with a sigmoid function called the output gate (o_t) which determines which part of C_t to use and a tanh activation function to determine the weight of the impact of C_t to be used. The resulting h_t is used as the output of the current time step, and will be transferred to the next time-step along with C_t for the process to be reiterated.

We created an LSTM with a single 10 neurons layer, with 6 time-steps. An Adam optimization method and a binary cross-entropy loss function were selected for our LSTM.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\hat{C} = tanh(W_{\hat{C}}[h_{t-1}, x_t] + b_{\hat{C}}) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C} \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

B. Hyperparameter Tuning

Most of the models we have presented and indeed most models in general offer built-in customization through so-called hyperparameters which can be adapted to fit the user's problem more adequately. This parameter could be for example the number of branches to construct when generating a Decision Tree or even the number of trees in a Random

Forest to give 2 simple examples. The idea being that there is not a ”one size fits all” configuration that is suitable for all applications but that each parameter of a model should be altered for better results. Various methods exist for finding these hyperparameters but we will be using 2 different types.

Grid Search: This is a brute-force search algorithm, which examines a manually specified subset of the hyperparameters[1]. The drawback of this approach is that the execution time is quite substantial, as it suffers from the curse of dimensionality.

Random Search: On the other hand, this approach examines hyper-parameter combinations in the specified subset randomly. This approach may be able to outperform the grid search, and it also allows to include parallelization in its tasks.

C. Sentiment Analysis

In order to integrate information derived from text into the aforementioned classification models, we first need to somehow transform the text into something the machine learning algorithms can assimilate. For this, we tried 2 different tools to extract the sentiment of our text.

VADER: Proposed by C.J. HUTTO & E.GILBERT (2014) [14], VADER is short for *Valence Aware Dictionary for sEntiment Reasoning*. This model is rule-based meaning that it is designed to focus on capturing certain grammatical features and lexicon-based meaning that is pre-trained on a database of lexical features. This database includes emoticons, acronyms and slang on top of more classical sentiment-carrying words[2] which it quantifies the sentiment of words between -4 and 4, -4 being the most negative possible sentiment, 0 being neutral and +4 being the most positive sentiment. Applying VADER to a sentence or paragraph outputs 3 probabilities one for the probability that the sentence is positive, one for it to be negative and one for it to be neutral. It also returns a “compound” number which regroups these 3 probabilities into a single number between -1 and 1 from most negative to most positive.

Its rich lexicon and specialized design allows it to work particularly well with social media posts. A useful feature for dealing with the more democratized subreddits.

TextBlob: From translation to spelling correction, this API offers a wide variety of NLP solutions. In our case, we are interested in its sentiment analysis functionality which is indeed very similar to the one offered by VADER. Although the process is not the same, the outcome is takes the same form were it not for the fact that TextBlob offers in addition to a sentiment rating between -1 and 1, a rating for the objectivity of the text, a feature which we are not interested in. The major difference lies in the database it is uses and that is slightly less advanced. We decided to use this sentiment analysis tool simply for comparison with VADER.

IV. Dataset Description

We were initially inspired by various projects found on internet[3] which we enriched substantially by adding more variables, taking a more sensible approach to Natural Language Processing (NLP) and using different target assets among other elements. Our observations span from mid 2014 to mid 2021 amounting to 1622 observations with pre-processing.
The data used in our prediction models can be categorized into 2 main types; numerical data which consist of prices and rates of various financial assets collected with the yfinance API, and text which we collect from titles of the top 25 posts of various subreddits using the Reddit API. The latter require more transformation.

The exhaustive description of all variables and the reason for their inclusion is described below.

A. Numerical Data

10y T-bill rate(TNX): This is the the yield received for investing in a US government issued treasury security with a maturity of 10 years. It is often used as the risk free rate, an essential component in financial modelling. It was included as a dependent variable in all our models with the objective to gauge the macroeconomic situation of the period.

The VIX index (VIX): This is the volatility index, sometimes dubbed the Fear Index, indicates the volatility anticipated by the market by extracting volatility from S&P 500 options for the 30 upcoming days. In other words, the VIX index measures the implied volatility embedded in market prices of the S&P500 index options. The Chicago Board Options Exchange (CBOE) is responsible of computing, publishing and also trading futures as well as options on the VIX. It is like the 10y T-bill rate, in order to capture a more short-term market sentiment.

Tesla Stock Price (TSLA): Used as a target variable in the form of a binary number taking the value 1 if there was an increase between the open and close price and 0 if not. Its lagged values are also used in all our models. The reason for including TSLA might seem arbitrary but it is in fact one of the archetypal “meme stocks” that Reddit communities such as r/WallStreetBets are greatly fond of. Hence, we expected a stronger predictive power with our sentiment analysis from Reddit posts.

S&P500 Index (GSPC): This equity index is one of the most prominent benchmarks used throughout the world. We included it as we expected news headlines to have an incidence on it. Similarly to TSLA, we used this index as a binary target variable as well as a dependent variable in its lagged form.

Bitcoin in USD: Included in order to have a non-equity asset and because it is a topic of discussion within many internet communities such as Reddit. It is also becoming a more prominent alternative to gold as a safe-haven and we therefore intended to also use it as a gauge of market confidence and expected its behaviour to differ substantially from the other assets which are equities our a collection of

[1] https://en.wikipedia.org/wiki/Hyperparameter_optimization (Accessed: 07/06/2021

[2] Source of lexicon: http://comp.social.gatech.edu/papers/ (Accessed: 07/06/2021

[3] https://www.kaggle.com/aaron7sun/stocknews (Accessed: 07/06/2021)

equities. As with TSLA and GSPC, we used this index as a binary target variable as well as a dependent variable in its lagged form.

B. Text Data

As previously stated, we gathered the titles of 25 most commented on posts for each of the following subreddits on each day.

r/Worldnews: This more formal subreddit sees posts which closely resemble those of a traditional news outlets. We believe it might contain information on events which impact financial markets.

r/Finance: Was included on the same grounds as the r/WorldNews but with information more material to our problem.

r/Cryptocurrency: Was added mainly to be used in relation with movements of the price of Bitcoin.

r/WallstreetBets: This infamous subreddit which was brought to the limelight due to the GameStop incident is an informal discussion forum used to share information and jokes on various financial topics. We included sentiment from this page as it often mentions Tesla and was expected to work well with TextBlob and Vader which are designed to function on more informal writing.

r/Investing: Included for many of the same reasons as WallstreetBets.

This leads to a total of 125 headlines gathered per day.

V. Implementation

A. Downloading the Data

To collect the top posts of the aforementioned subreddits which are relevant for our research, we had to develop an ”application” from the old Reddit to obtain the credentials to access live reddit activities and then we used a combination of two wrappers of pushshift.io, the reddit API which are called PRAW[4] and PSAW[5]. PRAW can rapidly collect up-to-date post at an optimized pace. However, the issue is that it cannot organize the output as we would like (i.e. top up-voted posts each day). Therefore, we used PSAW, as it was able to collect the up-to-date posts from PRAW and classify them as we would like. Consequently, we had to layer 2 wrappers to use the original API the way we wanted to. Nevertheless, the drawback of PSAW API was its execution time, as it requires a fairly significant amount of time to collect all top posts from 2014 until May 2021, and there was an issue when collecting the top posts based on their up-votes. We solved this issue by collecting the top posts based on their number of comments, which be believe is a good proxy for their number popularity. Afterwards, to get the time series for the S&P500, Tesla, Bitcoin, VIX index and 10y T-Bills, we used the unofficial Yahoo Finance ”yfinance” API, since the official one was decommissioned[6].

B. Data Pre-Processing

Having gathered the titles, we then pass each individual one through Vader to extract their sentiment as a numerical value between -1 and 1 and computed the average sentiment over the 25 headlines for each subreddit. These 5 averages are then used as a feature to try to predict the movement of the asset at hand. As a safety, we do the same operations using TextBlob instead of Vader and to be able to discern the effect of the sentiment analysis we also run our models without any of the information from the Reddit headlines.

Given that Bitcoin and Tesla are highly discussed on Reddit, we also created two additional variables, one that determines how many time Tesla or Elon Musk have been mentioned in the collected headlines and another for references to Bitcoin. Thus, we have in total of 12 variables originating from headlines, 5 sentiments using VADER, using TextBlob and 2 variables that determine the frequency with which Tesla/Elon Musk and Bitcoin have been mentioned.

The numerical data also underwent some modifications albeit much less transformative. We determine three binary columns that would be equal to 1 for a given day if the close price is higher than the open price (i.e. the price of the asset increased) for the S&P500, Tesla and BTC-USD, to be used as target variables for our predictions. We also use the log of these 3 time series as features in their lagged form as correlation among our assets might be a potential driver for forecasting.

Finally, all our variables, except the binary variables have been standardized by removing the mean and standard deviation. Hence, this leads to 3 different sets of feature variables. The first is with the Vader sentiment analysis and the other variables. The second is with the TextBlob sentiment analysis and the other variables. The third is without any sentiment analysis and only with the other variables. We also have 3 different possible target variables, the price movement of Tesla, the price movement of the S&P500 and the price movement of Bitcoin. This leads to us having 9 different combinations of variables to be used in the models.

Before applying our models, we split our data into an in-sample portion for training (data prior to January 1st 2020) and out-sample for testing (data after January 1st 2020 and until May 1st 2021) leading approximately to an 80/20 split. The rationale of doing it this way is because we will compare a hold-only position against our long/short position based on our predictions. We selected this splitting date to avoid the impact of the COVID pandemic on financial markets when training our models, which was indeed a major systematic event.

C. Data Analysis

To analyse some of our data, we plotted the cumulative returns and the average percentage of price increase over the full time frame of our assets, and the evolution of the VIX index and 10y T-bill rate. The relevant figures are figure 4 to figure 7 which we will comment on in order of appearance.

The first element to notice it that the out-sample performance of the 3 assets and the 2 macroeconomic indicators are

[4] https://github.com/praw-dev/praw(Accessed: 07/06/2021)

[5] https://github.com/dmarx/psaw (Accessed: 07/06/2021)

[6] https://github.com/ranaroussi/yfinance (Accessed: 07/06/2021)

extraordinary given the pandemic. This could be problematic since our models are trained given the behaviour of our variables in-sample and if that behaviour changes fundamentally, the models will not be able to adequately describe the data.

Regarding the returns of our 3 assets, we notice that for 2 of them (i.e. S&P500 Index and Bitcoin), there is a noticeable difference in the frequency of higher closes on the day versus lower ones. This could be an issue given that we might be lead to predict an upwards movement as our base-case when there would be no reason grounded in logic to do so. For example, had we used accuracy as our comparison metric, we would possibly always predict an upward movement which would be correct in the majority of cases. We also observe a high correlation in absolute terms between the equity-based time-series and the VIX which is normal given that the VIX is directly linked to them.

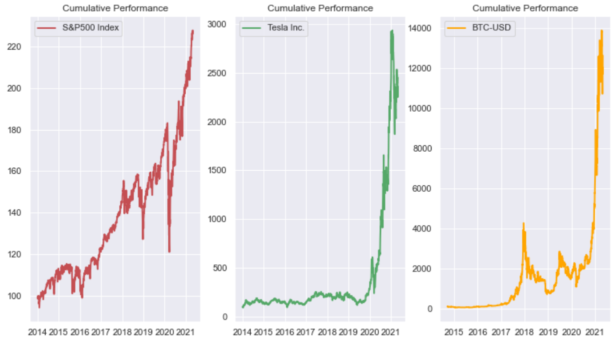

Fig. 4. Cumulative Returns of Each Assets

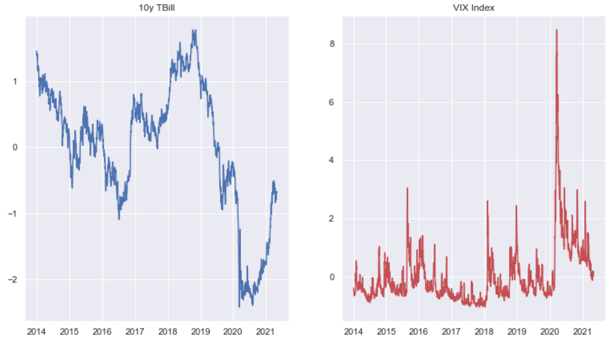

Fig. 5. Evolution of 10y T-Bill & VIX Index

The reason that explain how could the past returns of an asset could explain the future return of another asset is due to their most recent high correlations. Indeed, although figure 4 does indicate a low correlation among all our assets (except between the S&P500 index and Tesla Inc.), these correlations have been determined across the entire time horizon of our data set (i.e. from 2014 until 2021). By observing the correlation matrix in the out-sample only, as depicted on figure 8, we notice that the S&P500 index, Tesla and Bitcoin have been relatively correlated, partly explained by the latest market trends.

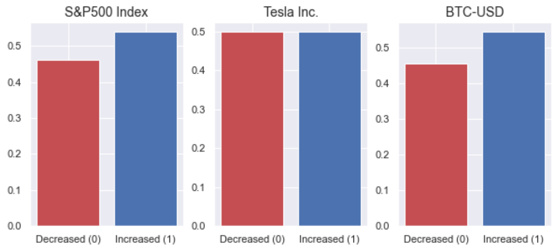

Fig. 6. Percentage of Increases & Decreases

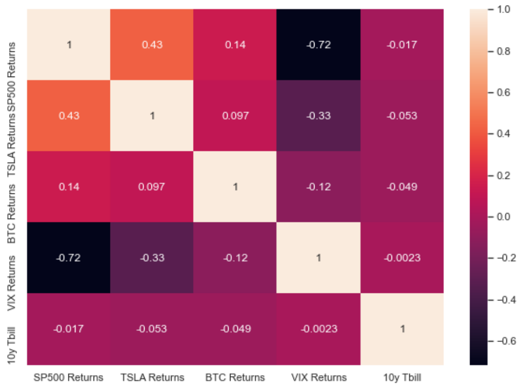

Fig. 7. Correlation Matrix (Entire Sample)

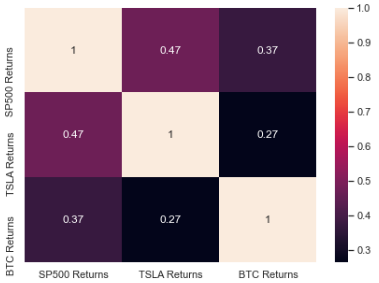

Fig. 8. Correlation Matrix (Out-Sample)

To understand the overall sentiments of our subreddits using VADER and Textblob, we determined the essential statistics to understand our inputs:

D. Model Implementation

To apply our models, we decided to take three different approaches regarding our hyperparameters. The first one was to not do any hyperparameter tuning and use the default ones. The second was to try some hyperparameter tuning by grid

TABLE I
DESCRIPTIVE TABLE OF TEXTBLOB SENTIMENT

	r/worldnews	r/finance	r/Crypto	r/wsb	r/investing
Count	2661	2661	2661	2616	2660
Cean	0.01578	0.06902	0.07566	0.03351	0.06373
STD	0.04166	0.05886	0.05442	0.06929	0.05009
Min	-0.14880	-0.14285	-0.31666	-0.33333	-0.10619
50%	0.01566	0.06556	0.07373	0.02796	0.06170
Max	0.16479	0.33456	0.57511	1.00000	0.31181

TABLE II
DESCRIPTIVE TABLE OF VADER SENTIMENT

	r/worldnews	r/finance	r/Crypto	r/wsb	r/investing
Count	2661	2661	2661	2616	2660
Mean	-0.19529	0.09474	0.06964	0.03710	0.06978
STD	0.09302	0.08022	0.07680	0.09877	0.06705
Min	-0.58370	-0.20408	-0.32353	-0.70030	-0.16017
50%	-0.19368	0.09383	0.06695	0.02971	0.07147
Max	0.13725	0.78450	0.39283	0.86580	0.33861

search. The third was to take the random search approach. For simplicity, hyperparameter tuning was performed on all our models, except the LSTM and DNN.

By the same token, we did three types of sentiment analysis. With TextBlob, with Vader since none have been particularly conceived for Reddit (as opposed to Twitter, since Vader is the go-to sentiment prediction model) and without any at all to see if there is any difference.

Given all the methods detailed in section III, and different variables given in section IV, running our code results in 180 different combinations, or 60 per model as detailed by equation (7).

$$(T * TM + U) * S = (6 * 3 + 2) * 3 = 60 \tag{7}$$

T	Tuned models
TM	Tuning methods
U	Untuned models
S	Sentiment extractors

However, there is no reason to be taken aback by this number as we automatically selected for each asset the combination with the highest accuracy measure. The reason for using accuracy as a the benchmark metric is because for our investment strategy, we long the asset when our model predicts an increase in its value and we short it when we expect it to go diminish. Therefore, we want to maximize the number of both true positives and true negatives.

Thereafter, we computed the performances of our long/short trades based on our predictions on whether the price will increase or decrease, compared to a hold-only position for each asset. Finally, we determined the annualized returns $r_{Annualized}$, annualized volatility $\sigma_{Annualized}$, Sharpe ratio, hit ratio and maximum drawdowns to evaluate its performances. The three latter are calculated with equation (8), equation (9) and equation (10).

$$\text{Sharpe Ratio} = \frac{r_{Annualized} - r_f}{\sigma_{Annualized}} \tag{8}$$

Where we assume a risk-free rate r_f of zero.

$$\text{Hit Ratio} = \frac{n_{pos}}{N} \tag{9}$$

Where n_{pos} is the number of times the portfolio witnessed a positive return and N is the total number of observations.

$$\text{Max Drawdown} = \min\left[\frac{P_s - P_t}{P_t}\right] \tag{10}$$

With $t < s$.

VI. RESULTS

After implementing our code, the results of our long/short position based on the models' predictions were interesting. Overall, we notice that they have been provide better returns than a hold-only, but not consistently. This could be due to many factors. For example, it is extremely difficult to determine an algorithm being able to outperform the benchmark consistently. Secondly, the scarcity of data may be a factor affecting our performances, as we were able to retrieve data only from 2014. The reason being that some subreddits and the price of Bitcoin lacked information before then. Nevertheless, under some settings, we have been able to outperform the hold-only benchmark. Among all our assets, Tesla seems to be the most difficult one to estimate, particularly at the end of the out-sample. This could be due to the large volatility coming from this security, and the recent market turmoils forged by the whole r/wallstreetbets drama. Regarding our prediction models, we notice that a more simple model can be more suitable than a complex one. Indeed, for example a random forest or a support vector machine seems to be more appropriate than a LSTM or a DNN under some settings. This could be due to our relatively low number of observations, since these highly complex models requires a large amount of observations to be more accurate. After applying our models using no hyperparameter tuning, grid search and random search, we obtained slightly different results. For the sake of being concise, we will present the models which generated the best results in terms of accuracy and cumulative returns.

A. S&P500 Index

Without Reddit Posts: When excluding Reddit posts into our input, we observe that the best model to predict the price movement of the S&P500 Index was a SVM without hyperparameter tuning, yielding an accuracy of 0.597. With this setting, we have been able to deliver better performances than a hold only position, as depicted on figure 9 and in the first two columns of table III. We notice that the deviation from the benchmark starts at the beginning of the out-sample in February 2020, as market sentiments were unstable amid the first cases of COVID in Europe. Thereafter, our model seems to struggle by predicting mainly prices increases, although March 2020 was a rough period for global markets.

With Reddit Posts: When including Reddit posts into our output, we notice that the best model to predict the price movement of the S&P500 Index was our deep neural network described on section III using the sentiment created by VADER, yielding an accuracy of 0.578. Again, we have been able to deliver better performances than a hold only position, as depicted in figure 10 and in the first and last columns of table III. Compared to the previous long/short portfolio, we notice that including the overall sentiment of subreddit posts does improve the performances of our portfolio in term of cumulative returns. Indeed, as the S&P500 is sensitive to global market trends, and is more conventional that Bitcoin and Tesla, we expected the addition of news headlines from r/WorldNews for example to hold some predictive power. Our results do indicate that including the sentiments of our aforementioned subreddits allow to improve the performances of our portfolio, as described on the last two columns of table III.

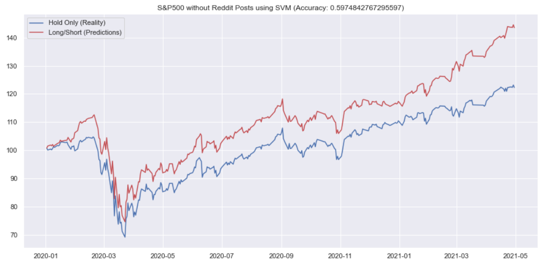

Fig. 9. S&P500 without Reddit Posts (SVM, no HPT)

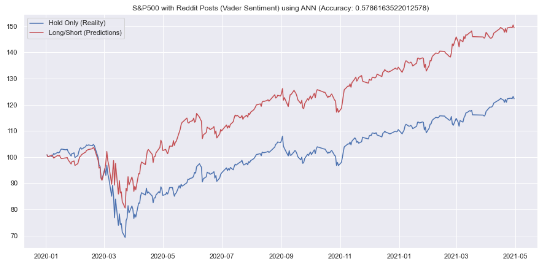

Fig. 10. S&P500 with Reddit Posts (Vader, DNN)

TABLE III
S&P500: HOLD-ONLY VS LONG/SHORT

	Hold-Only	No Reddit Posts	Reddit Posts
Returns	0.209084	0.335707	0.367639
Volatility	0.313420	0.312983	0.312840
Sharpe Ratio	0.667104	1.072604	1.175166
Max Drawdown	-0.339250	-0.336719	-0.223359
Hit Ratio	0.569182	0.613208	0.581761

B. *Tesla Corporation*

Without Reddit Posts: When excluding Reddit posts from our input, we observe that the best model to predict the price movement of Tesla Corp. was an adaptive boosting with grid search hyper-parameter tuning, yielding a low accuracy of 0.509. Compared to predicting the S&P500 Index, we observe the difficulty of predicting the price movement of TSLA, since we have not been able to deliver better performances than a hold-only, as depicted on figure 11 and in the first two columns of table IV. This could be due to the fact that the S&P500 can be more sensitive to more conventional market measures such that market volatility and interest rates, captured by the VIX index and the 10y T-bills in our model. On the other hand, TSLA is more subject to idiosyncratic factors (i.e. business operations, profitability, etc.) in which our model is not able to capture.

With Reddit Posts: When including Reddit posts into our input, we observe that the best model to predict prices movement of Tesla Corp. was again our deep neural network including the sentiments created by VADER, yielding an accuracy of 0.537. We notice the complete failure of predicting the price movement of Tesla, as we have not been able to generate better performances than the hold-only portfolio. This is because the model failed to recognize the correct result when it really mattered, (i.e. between November 2020 to January 2021) when the stock saw a meteoric rise due to the fact that we concentrate on predicting the direction of the movement of our asset. Our models could not be trained to predict such staggering rise in prices. This in no way tells us about the magnitude of these movements.

Furthermore, the the extreme volatility of TSLA exhibited in the out-sample period is unlike what the model was trained on as the annualized standard deviation is a staggering 86% compared to 45% during the in-sample period.

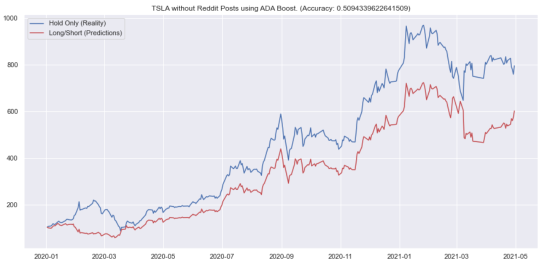

Fig. 11. TSLA without Reddit Posts (ADA Boost., Grid Search)

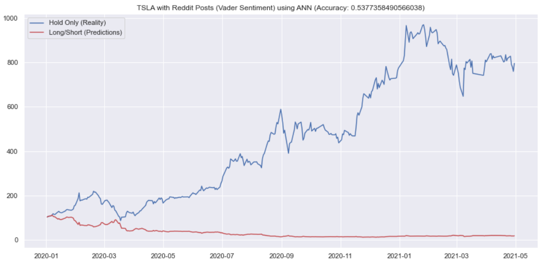

Fig. 12. TSLA with Reddit Posts (Vader, RF, no HPT)

TABLE IV
TSLA: HOLD-ONLY VS LONG/SHORT

	Hold-Only	No Reddit Posts	Reddit Posts
Returns	2.020214	1.802307	-0.958180
Volatility	0.862006	0.863921	0.869256
Sharpe Ratio	2.343619	2.086194	-1.102298
Max Drawdown	-0.606265	-0.516082	-0.893566
Hit Ratio	0.556604	0.550314	0.490566

C. Bitcoin

Without Reddit Posts: When excluding Reddit posts from our input, we observe that the best model to predict the price movement of Bitcoin was our Deep Neural Network described on section III yielding an accuracy of 0.597. Under this configuration, we observe on figure 13 and table V that our long/short position has been able to provide better results that a hold-only position. Since Bitcoin is a ”peer-to-peer electronic cash system”[7], much of the factors affecting its price are systematic, which can explain our great performances based only on market volatility, interest rates, and past returns of itself, TSLA and the S&P500 index.

With Reddit Posts: When including Reddit posts in our input, we observe that the best model to predict the price movement of Bitcoin was a logistic regression with grid search hyper-parameter tuning and using sentiment created by Textblob, yielding an accuracy of 0.584. As depicted on figure 14 and table V, we notice that we have been able to deliver better results than a hold-only position. Nevertheless, by adding the sentiments of Reddit posts, we failed to outperform the previous long/short portfolio which excludes the sentiment. This result can be partly explained by the difficulty of interpreting comments/posts that may affect the price of Bitcoin. Indeed, as crypto-currencies seems to be more appealing to younger generations, the discussions and posts surrounding these assets take the form of ’memes’ and slang which are not well assimilated by our sentiment analysis methods. For example when we observe a common comment such as ”Bitcoin to the moon”[8], which does not make any sens a priori, but does indeed expresses enthusiasm for the asset and an expectation of dizzying heights. Another example was the famous tweet of Tesla’s CEO, Elon Musk, writing on the 19th of May 2021 that ”Tesla has diamond hands”[9] while using emojis for diamond hands, following Tesla’s decision of purchasing a massive amount of the crypto-currency. This tweet led to a considerable volume of trades on BTC and hence a sizeable increase in price but was not assimilated by our model in the slightest given that this sentence yielded a score of 0.0 - neutral in both of VADER and TextBlob.

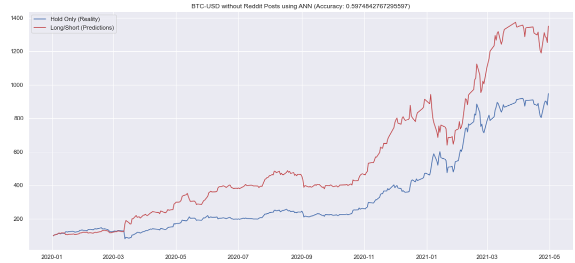

Fig. 13. BTCUSD without Reddit Posts (DNN)

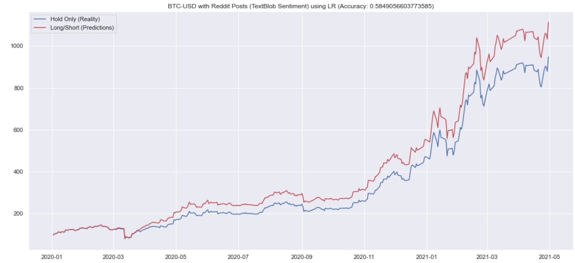

Fig. 14. BTCUSD with Reddit Posts (Textblob, LR, Grid Search)

TABLE V
BTCUSD: HOLD-ONLY VS LONG/SHORT

	Hold-Only	No Reddit Posts	Reddit Posts
Returns	2.051713	2.304001	2.179307
Volatility	0.705362	0.702264	0.703841
Sharpe Ratio	2.908740	3.280820	3.096305
Max Drawdown	-0.454874	-0.321914	-0.444715
Hit Ratio	0.575472	0.591195	0.584906

D. Key Takeaways

By implementing 60 different models per target, we cast a relatively large net which would inevitably allow us to boast some successful models. We notice that our DNN, a model on the more complex side of the spectrum was able to provide the best performance for 2 out of 3 of our assets but our most complex one, the LSTM, which we hoped would work best since it is designed for applications such as this one, was nowhere to be seen due, we believe, to the lack of observations.

We also noticed mixed results with our sentiment analysis but we have not lost hope in its potential and see where we could better approach our NLP. Moving forward, we would probably focus more on certain keywords, and perhaps even analyse the comments of the posts instead of just their titles to understand how the communities feel about the subject and not just the sentiment of the person submitting the post.

It is important to note that given that our out-sample period was abnormally turbulent for financial assets. Therefore, we cannot assert that the models which performed well will continue to do so if we were to pursue the strategy they dictate. We do however see great potential in using them as a starting point for for a more fine-tuned strategy.

[7] https://bitcoin.org/bitcoin.pdf (Accessed: 06/06/2021)

[8] https://www.forbes.com/sites/kenrapoza/2021/02/21/bitcoin-to-the-moon-is-it-worth-chasing-the-crypto-bull-market/?sh=2d36eed6446d (Accessed: 06/06/2021)

[9] https://www.cnbc.com/2021/05/19/elon-musk-tweets-diamond-hands-emoji-amid-bitcoin-drop-implying-tesla-wont-sell.html (Accessed: 06/06/2021)

VII. Conclusion

The r/wallstreetbets community has been arguably one of the more influential factors affecting markets in 2021 so far. As many young retail investors gathered together online, aiming to disturb the hierarchical powers in financial markets, by bringing notorious hedge funds to their knees in a ”power to the people” [10] movement. This event was both spectacular, by its sheer amplitude and impact to global markets, but also very frightening for the stability and credibility of financial markets which should in theory reflect the fair value of a company or asset. Thus, raising many questions. What if this becomes a common occurrence? Will it be seen as market manipulation and punishable by the Security and Exchange Commission (SEC)? And if so who would be at fault since the event was created by an entire community? Was this debacle a testament to the people's disdain for the financial industry and a warning or simply a cash grab driven by greed? As the legendary investor Warren Buffet once said, ”be fearful when others are greedy, and greedy when others are fearful[11]”.

In hopes of capitalizing on Reddit's newfound celebrity and credibility, we attempted to leverage sentiment analysis performed on subreddits, coupled with historical financial times series, to apply machine learning methods from simpler ones to the a priori more specialized ones such as LSTM, to predict whether the price of the S&P500 index, Tesla Corp. and Bitcoin will increase or decrease. Consequently, we've been able to create a long/short position on each asset based on our predictions, and we compared it to a hold-only position.

Using various models as described on section III and select the best predictions based on accuracy, we notice that overall when excluding the reddit sentiment, our long/short positions has outperformed the hold-only position for the S&P500 index and Bitcoin, but not for TSLA. The reason why we obtained better results for the index and bitcoin is that they are less likely to be affected by idiosyncratic factors but rather almost systematic factors. On the other hand Tesla is still very likely to be affected by idiosyncratic factors, which our data could not capture. Our results did improve when including the sentiment of Reddit posts in the case of S&P500. Nevertheless, our results did not improve for Bitcoin and Tesla, and sometimes did worsen the predictions. This could be due to the difficulty of VADER and Textblob to understand new jargon of recent years.

To conclude, Natural Language Processing for event-based trading is an exciting and promising aspect that need to be investigated further as an add-in for any investing algorithm. Although we acknowledge the weakness of our models, particularly the difficulty of interpreting the sentiment of a Reddit posts, we believe there is room for improvements. For example, we could supplement TextBlob and VADER's lexicons with a better adapted vocabulary. We would also envisage more focused and targeted approach when collecting posts to predict movements in a particular company as we believe one of the pitfalls of our data selection was that it suffered from noise originating from irrelevant information. We could combat this by inputting only posts mentioning keywords in connection with the asset in question and also selecting subreddits which are more focused on the particular target. The reason for us not doing this from the start is the lack of high rated posts to select from. Another potential improvement could be to dynamically include new stocks or investment products when they are becoming more popular, for example AMC Entertainment as soon as some new signals are detected in r/wallstreetbets. Our project is merely an exploration on this fascinating sentiment-based trading approach.

References

[1] Malkiel, B.G. and Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. The journal of Finance, 25(2), pp.383-417.

[2] Malkiel, B.G., 2003. The efficient market hypothesis and its critics. Journal of economic perspectives, 17(1), pp.59-82.

[3] Oberlechner, T. and Hocking, S., 2004. Information sources, news, and rumors in financial markets: Insights into the foreign exchange market. Journal of economic psychology, 25(3), pp.407-424.

[4] Leinweber, D. and Sisk, J., 2011. Event-driven trading and the “new news”. The Journal of Portfolio Management, 38(1), pp.110-124.

[5] Mehtab, S. and Sen, J., 2020. Stock price prediction using convolutional neural networks on a multivariate timeseries. arXiv preprint arXiv:2001.09769.

[6] Akita, R., Yoshihara, A., Matsubara, T. and Uehara, K., 2016, June. Deep learning for stock prediction using numerical and textual information. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) (pp. 1-6). IEEE.

[7] Tsai, C.F. and Hsiao, Y.C., 2010. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. Decision Support Systems, 50(1), pp.258-269.

[8] Turing, A.M., 1950. Mind. Mind, 59(236), pp.433-460.

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546.

[10] Jermann, M., 2017. Predicting Stock Movement through Executive Tweets.

[11] Wai, B. and Peng, C., 2018, CS 230 Final Report: Predicting US Stock Market Movement from Political Tweets.

[12] Hu, Z., Zhao, Y. and Khushi, M., 2021. A survey of forex and stock price prediction using deep learning. Applied System Innovation, 4(1), p.9.

[13] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[14] Hutto, C. and Gilbert, E., 2014, May. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 8, No. 1).

[10] https://medium.com/swlh/gamestonks-power-to-the-people-1e0d2b1f4ee4(Accessed: 06/06/2021)

[11] https://www.investopedia.com/articles/investing/012116/warren-buffett-be-fearful-when-others-are-greedy.asp(Accessed: 07/07/2021)