Cryptocurrency Returns Predictor

General Purpose

Study whether several measures of investor sentiment in the cryptocurrency market could predict the returns of this market. Also compare the performance of 3 different models (two of which are *Random Forest)* in predicting cryptocurrency returns.

Original Dataset

Stored in the *"final-dataset.csv"*. The dataset is collected using multiple methods between the 28 November 2014 to 25 July 2020 (2041 daily observations). The variables and their sources are described as in Table 1.

Table 1: Variable description and data sources. Source: Own elaboration

Variable	**Description**	**Source**
CRIX	A capitalization-weighted market index for cryptocurrencies (analogous to the S&P500 of the U.S stock market, or the DAX of the German stock market)	Constructed by Trimborn & Härdle (2018). Retrieved at: http://data.thecrix.de/data/crix.json

$SENT_{StockTwits}$	Daily sentiment score of StockTwits messages strictly related to cryptocurrencies (containing 1 of 532 crypto tickers supported by the platform)	Messages retrieved using StockTwits public API, converted into unigrams and bigrams, then graded their sentiment using the lexicon[1] created by Chen et al. (2019).
$SENT_{RedSub}$	Daily sentiment score of all submissions in the (only) two subreddits related to cryptocurrency with more than 1 million subscribers (r/Bitcoin and r/CryptoCurrency)	Textual data on Reddit retrieved using Reddit Pushshift API, also converted into unigrams and bigrams, then graded their sentiment using the lexicon[2] created by Chen et al. (2019).
$SENT_{RedCmt}$	Daily sentiment score of all comments in the (only) two subreddits related to cryptocurrency with more than 1 million subscribers (r/Bitcoin and r/CryptoCurrency)	
VCRIX	A cryptocurrency market volatility index (similar to the VIX of the U.S. stock market, or the VDAX of the German stock market)	Created by Kolesnikova (2018) based on the CRIX index by Trimborn & Härdle (2018).
VOL_{Trade}	Daily market transaction volume, quoted in USD	Retrieved using the Nomics Public API.

[1] Could be found at the main author's personal page: https://sites.google.com/site/professorcathychen/resume

[2] Could be found at the main author's personal page: https://sites.google.com/site/professorcathychen/resume

VOL_{Google}	Daily Google search volume	Computed by Google Trends. Retrieved using the Pytrends package.
$VOL_{StockTwits}$	Daily message volume on StockTwits	StockTwits public API
VOL_{RedSub}	Daily submission volume on Reddit	Reddit Pushshift API
VOL_{RedCmt}	Daily comment volume on Reddit	

Models Description

For the purpose of *Random Forest* application, both *Random Forest Regressor* and *Random Forest Classifier* will be implemented to predict the cryptocurrency market returns (denoted as *RM*, formula given in equation 1). For the purpose of performance comparison, a *Vector Autoregression VAR(p)* model will also be applied (Goodness-of-fit criteria *BIC* suggests $p = 5$).

$$RM_t = \frac{CRIX_t \ - \ CRIX_{t-1}}{CRIX_{t-1}} \tag{1}$$

More specifically, *Random Forest Regressor* is utilized to predict the exact values of future returns, while *Random Forest Classifier* is only for forecasting future price directions (i.e., will the next day returns are positive or negative?). A drawback of differencing is that we have to drop the first observation of our dataset (which is on 28 November 2014).

On the one hand, the features chosen for *Random Forest* models are lagged values of all 9 sentiment measures, at lags 1-5 (to match the number of lagged values used by the *VAR* model). Thus, a modified matrix of features to be used for *Random Forest* could be seen as displayed in Table 2.

Table 2: Examples of lagged sentiment measures (at lags 1-5). Source: Own elaboration

Date	$SENT_{StockTwits(-1)}$	...	$VOL_{RedSub(-5)}$	$VOL_{RedCmt(-5)}$
04-12-2014	-0.176	...	4	388
05-12-2014	0.019	...	2	256
...	...	...	...	...
24-07-2020	0.252	...	1003	370
25-07-2020	0.247	...	806	458

On the other hand, the independent variable in the *VAR(5)* model is the lagged values of the first principal component[3] of all 9 sentiment variables, which is called the *composite sentiment index* (denoted as *SENT*). The component loadings are given as in equation 2.

$$\begin{aligned} SENT = \; & 0.116SENT_{StockTwits} + 0.166SENT_{RedSub} + 0.226SENT_{RedCmt} + 0.207VCRIX \\ & + 0.162VOL_{Trade} + 0.460VOL_{Google} + 0.394VOL_{StockTwits} + 0.484VOL_{RedSub} \\ & + 0.487VOL_{RedCmt} \end{aligned} \tag{2}$$

For all three models, the training data starts from 29 November 2014 to 04 March 2019 (75% dataset), and the test data starts from 05 March 2019 to 25 July 2020 (25% dataset).

[3] This approach is adopted from Brown & Cliff (2004) and Baker & Wurgler (2007), two highly-cited studies about investor sentiment in the stock market. The idea is to capture the maximum joint-variation among the individual sentiment measures, since these authors believed those individual measures are often highly correlated. More importantly, Baker & Wurgler (2006) insists that it is nearly impossible for imperfect proxies to stay useful over time. In other words, while some sentiment proxies measure properly at a point in time, the others may only become valid at another time. Thus, for empirical experiments in the long horizons, it is sensible to combine a bunch of available proxies into a composite sentiment index that might have the potential to remain effective for a prolonged duration.

Random Forest Regressor Result

The hyper-parameters are chosen as: the number of trees is fixed at 500, the number of features to consider when looking for the best split is 7 (rounded square of 45 - total number of features).

Results show that while predicting returns during the test period, the *Random Forest Regressor* model has a *Mean Squared Forecast Error (MSFE)* = 0.00154. In terms of predicting the direction of price, the confusion matrices for training data and test data are given in Table 3. It could be seen that this model predicts correctly only 56.19%.

Table 3: *Random Forest Regressor*'s Confusion Matrices. Source: Own elaboration

(a) Test Data

		Actual Classes		
		Negative	Neutral	Positive
Predicted Class	Negative	222	0	20
	Neutral	1	0	0
	Positive	202	0	64

(b) Train Data

		Actual Classes		
		Negative	Neutral	Positive
Predicted Class	Negative	596	0	68
	Neutral	1	0	2
	Positive	31	0	829

Random Forest Classifier Result

The hyper-parameters are the same as the previous model. The only difference is that instead of predicting the exact value of returns, we only predict the price directions in this model.

The confusion matrices for training and test data are given in Table 4. The prediction accuracy in test set is significantly better than the *Random Forest Regressor*, at around 61.89%. However, the confusion matrix for training data indicates signs of over-fitting the data.

Table 4: *Random Forest Classifier*'s Confusion Matrices. Source: Own elaboration

(a) Test Data

		Actual		
		Negative	Unchanged	Positive
Predicted	Negative	145	0	97
	Unchanged	1	0	0
	Positive	96	0	170

(b) Train Data

		Actual		
		Negative	Unchanged	Positive
Predicted	Negative	664	0	0
	Unchanged	0	3	0
	Positive	0	0	860

VAR(5) Model Result

Stationary tests (KPSS and ADF) show that the market return (*RM*) series is stationary, however the sentiment index (*SENT*) is non-stationary I(1) process. Thus, in our *VAR* model, we regress *RM* on the first difference of the sentiment index (or $\Delta SENT$). Table 5 reports the result from estimating the *VAR* model using $\Delta SENT$ to explain *RM*. Using the *VAR* model to predict returns during the test period receives a Mean Squared Forecast Error (MSFE) = 0.00148 (slightly smaller than the *Random Forest Regressor*). Note that future returns could be predicted using the regression equations of the *VAR* model, given formally in equation 3.

$$\widehat{RM}_t = \widehat{\propto} + \sum_{i=1}^{5} \hat{\beta}_i \Delta SENT_{t-i} + \sum_{i=1}^{5} \hat{\delta}_i RM_{t-i} \tag{3}$$

where the constant $\widehat{\propto} = 0.002442$ and other lagged coefficients of $\Delta SENT$ and *RM* ($\hat{\beta}_i$ and $\hat{\delta}_i$, respectively) could be found in table 5.

In terms of predicting the direction of price, the confusion matrices for test data of this model are given in Table 6. The model predicts correctly only 54.62% during the test period.

Table 5: VAR(5) Results (RM & $\Delta SENT$). Source: Own elaboration

Independent Variable	**Lag**	**Dependent Variable**	
		RM	$\Delta SENT$
RM	1	-0.0342	0.7980***
	2	0.0169	0.6393***
	3	0.0355	0.4143
	4	0.0093	1.6390***
	5	0.0110	0.4316
$\Delta SENT$	1	0.0046***	-0.2284***
	2	0.0004	-0.3444***
	3	0.0028	-0.2665***
	4	0.0042**	-0.2235***
	5	0.0029*	-0.2179***

* Indicate significance at the 10% level
** Indicate significance at the 5% level
*** Indicate significance at the 1% level

Table 6: *VAR(5)* Confusion Matrix (Test set). Source: Own elaboration

		Actual		
		Negative	Unchanged	Positive
Predicted	Negative	57	0	185
	Unchanged	0	0	1
	Positive	45	0	221

Model Performance Comparison

It appears that the *Random Forest Classifier* seems to perform the best at predicting future directions of the cryptocurrency market while the worst belongs to the *VAR(5)* time series model. Now we simulate trading strategies that are based on the returns predicted by those models to see which model produces the most profitable signals. We will also compare those strategies to the classic one of *Buy-and-hold* the market index (*CRIX*) to see if we could outperform the market. The rule to generate trading signals is fairly simple, in which we go long ($BUY = 1$) when the forecasted return is greater than 0, go short ($BUY =$ -1) when the forecasted return is less than 0, and wait (do nothing) otherwise. In mathematical terms, this could be expressed as:

$$BUY_t = \begin{cases} 1, & if\ \widehat{RM}_t > 0 \\ -1, & if\ \widehat{RM}_t\ < 0 \\ 0, & otherwise \end{cases} \tag{4}$$

Assuming transaction costs are negligible, the cumulative return of a strategy at day t is given by:

$$R_t^{Strat} = \prod_{i=1}^{t}(BUY_i * RM_i\ \ +\ 1) - 1 \tag{5}$$

The cumulative returns of all strategies are plotted agaisnt each other in Figure 1. An interactive version of the figure could be found in the file *"strats.html"*. As expected, the strategy based on the *Random Forest Classifier* performs the best as it tops out an astonishing daily return of ~91bps (which is around 4.79 times of the daily returns generated by the *Buy-and-Hold* strategy). Interestingly, the strategy based on the *VAR(5)* model (~48bps) significantly outperforms the one based on *Random Forest Regressor* (~19bps, which is just very slightly better than the holding the index) although the earlier has fewer times of predicting correctly than the latter.

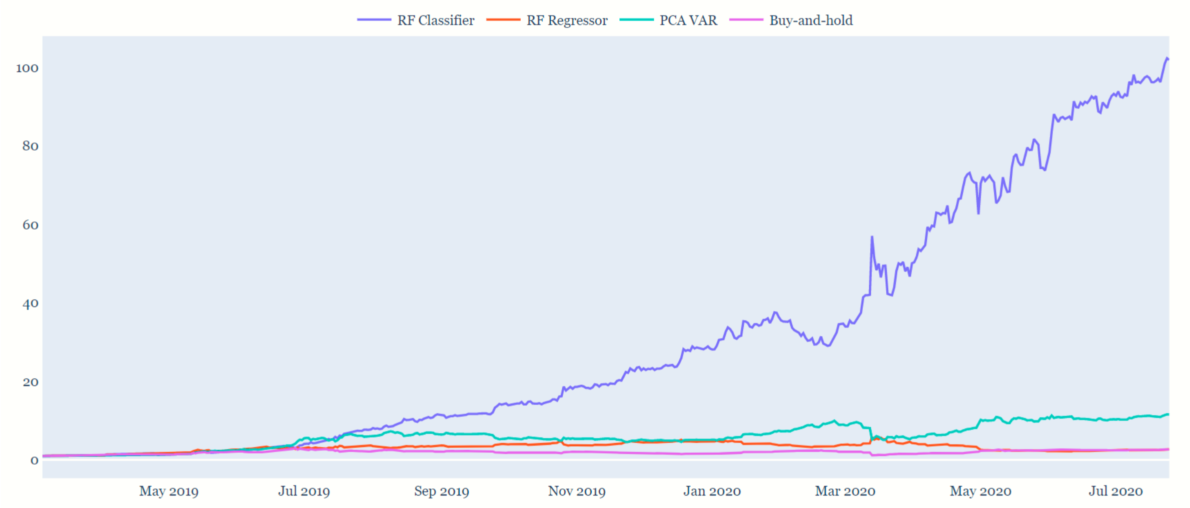

Figure 1: Trading strategies based on different models. Source: Own elaboration.

References

Baker, M., & Wurgler, J. (2006). Investor Sentiment and the Cross-section of Stock Returns. *Journal of Finance*, *61*(4), 1645–1680. https://doi.org/10.1111/j.1540-6261.2006.00885.x

Baker, M., & Wurgler, J. (2007). Investor Sentiment in the Stock Market. In *NBER Working Paper* (Issue No. 13189). http://www.nber.org/papers/w13189

Brown, G. W., & Cliff, M. T. (2004). Investor Sentiment and the Near-term Stock Market. *Journal of Empirical Finance*, *11*(1), 1–27. https://doi.org/10.1016/j.jempfin.2002.12.001

Chen, C., Despres, R., Guo, L., & Renault, T. (2019). What Makes Cryptocurrencies Special? Investor Sentiment and Return Predictability During the Bubble. *SSRN Electronic Journal*, 1–36. https://doi.org/10.2139/ssrn.3398423

Kolesnikova, A. (2018). *VCRIX - Volatility Index for Cryptocurrencies on the Basis of CRIX*. https://edoc.hu-berlin.de/bitstream/handle/18452/20056/master_kolesnikova_alisa.pdf?sequence=3&isAllowed=y

Trimborn, S., & Härdle, W. K. (2018). CRIX an Index for cryptocurrencies. *Journal of Empirical Finance*, *49*(October), 107–122. https://doi.org/10.1016/j.jempfin.2018.08.004