Bitcoin Stochastic Stock to Flow Model*

Predicting the Bitcoin price with Ridge Regression

Nathaniel Fowler Philippe Estines

June 12, 2020

Abstract

Many attempts have been made to predict the price and market capitalization of Bitcoin - with varying degrees of success. Machine learning models, such as Ridge Regression, allow researchers to optimize a model for predictions via a process called regularization. In this paper, we aim to make the following contributions: (1) further research on what drives the price of Bitcoin, (2) build a low root mean squared error Ridge Regression model using only Bitcoin market-specific factors and investor attention, and (3) display the superior predictive power of the Ridge model.

*Special thanks to Katya Vasilaky, Ph.D. for assistance with Ridge in Python, and to Nick Emblow for assistance with constructing the stock-to-flow variable in Stata

1 Introduction

Bitcoin, the worlds largest cryptocurrency, is a provably scarce asset. Bitcoin has a fixed supply of 21 million, which is released on a supply schedule that is decreasing over time (Nakamoto, 2008). Bitcoin is a unique financial asset, in that it creates and stores data about the protocol on an immutable database called the blockchain - data which will henceforth be referred to as "on-chain" data. On-chain data is of interest because anybody who runs the Bitcoin code can verify this data, and it cannot be edited. We will use a combination of on-chain data and investor attention metrics in our analysis.

Bitcoin is of interest for a myriad of reasons, but perhaps most notable to Economists would be its meteoric rise from the aftermath of the Great Recession. Bitcoin was the best-performing asset of the 2010's, gaining some 9,000,000%. As such, an accurate pricing model would prove invaluable for any investor or academic alike who wishes to determine whether such gains can continue into the future. This paper will highlight the existing literature on Bitcoin pricing models, and will use modern machine learning techniques to create a unique pricing model with a distinct advantage in prediction compared to existing models.

1.1 Literature Review

Aiming to quantify the scarcity of Bitcoin, we began our research on the stock-to-flow ratio, which is a metric for measuring the scarcity of rare assets such as precious metals. Adrangi et al. (2015) finds that the flow environment (often measured as the annual inflation rate) allows for supply and demand pressures to determine commodity prices. Conversely, the stock-to-flow environment depends on investor speculation of the supply and demand for inventories. The authors examine how to classify gold, silver and copper in this way. This relates to cryptocurrency because the method in which Bitcoin is created is similar to that of mining precious metals. Like precious metals,

scarcity is one of the reasons that Bitcoin has value, as the production of all of these assets is difficult and requires specialized equipment, which separates them from other commodities. Additionally, precious metals and Bitcoin are both speculative financial assets (in that they do not generate a return on their own), which justifies the use of the stock-to-flow ratio in analysis.

Despite the comparison to precious metals, which have a long history of being used as a stable store of value, Bitcoin is one of the most volatile assets in human history and has been used as a store of value for at most a decade. As such, there is more to consider than just the stock-to-flow ratio when analyzing the price fluctuations. Liu and Tsyvinski (2018) takes a comprehensive look at cryptocurrencies, examining how they behave over time and what factors influence price. The authors find that Bitcoin does not behave like a currency, commodity, or precious metal and has no exposure to most common stock market and macroeconomic factors. Likewise, the returns of currencies and other commodities have no statistically significant relationship with Bitcoin returns. Instead, Bitcoin market-specific factors such as momentum and investor attention were the only statistically significant predictors that the authors identified. The paper establishes strong evidence of present returns predicting future returns, both on a daily and weekly basis. In terms of investor attention, there was at least a first week statistically significant increase in returns after a one standard deviation increase in the current week's searches, ranging from about 2% to 11%.

Liu and Tsyvinski (2018) construct a metric for investor sentiment that takes the ratio of the volume for the search term "Bitcoin" to the search term "Bitcoin hack". They found that a higher relative volume of searches for "Bitcoin hack" negatively and statistically significantly predicts the next 5 weeks of Bitcoin returns. The authors also found that traditional asset pricing models such as CAPM and Fama French 5-factor fail to comprehensively and statistically significantly model the price or market cap of Bitcoin. The purpose of this paper was not to predict the price or market capitalization of Bitcoin,

yet their framework for investor sentiment and failure to adapt existing asset pricing models to Bitcoin provided insights on how to create a more refined model.

Goczek and Skliarov (2019) tests a variety of influences on the Bitcoin price. The authors review data from nine other papers and create a factor augmented vector error correction model (FECM), an extension of the factor augmented vector autoregressive approach (FAVAR) which is typically used in monetary policy research (see Bernanke et al. (2005) for example). This allows for factors to be determined from the twenty-three chosen variables to be used in regression. One must also note that these variables were actually grouped into topics such as global economic climate, attractiveness for investors and stock/commodity market conditions. The authors also use bootstrapping to determine standard errors. Interestingly, they find that there is no direct influence of the global economic climate on price, concluding that Bitcoin cannot be depended on as insurance for global crises. However, this does mean that those events do not necessarily impact the price of Bitcoin as much as other asset classes. Ultimately, the paper finds that attractiveness to investors is the main predictor for the price of Bitcoin, another indication that investor attention and attractiveness is key to the regression.

Furthering the research on investor attention, we next examine the viability of using Twitter data as a predictor variable. Abraham et al. (2018) tested the viability of using Google Trends and Twitter data to predict Bitcoin price with mixed results. While sentiment did not provide statistically significant results, the authors did find that tweet volume was a statistically significant predictor of price direction. Additionally, the Google Trends data was statistically significant enough to include in their regression. Furthermore, Ranco et al. (2015) studied the effects of Twitter sentiment on stock price returns. Similar to the previous example, sentiment alone was statistically insignificant. However, when the volume of tweets peak, the prevailing sentiment did imply the direction of returns. Thus, there is some potential and justification for using similar analysis on Bitcoin.

To date there are no academic papers that utilize the stock-to-flow ratio to predict

the price of Bitcoin (with most research on this model found in industry papers, not in academic circles). Additionally, mining difficulty has not been investigated as a potential explanatory variable for predicting price in an academic setting. The relationship has been hinted at, however, such as in Taylor (2013), which shows figures suggesting there exists a correlation between the Bitcoin price and mining difficulty, though no formal analyses were performed. Thus, this paper aims to contribute to the existing literature by building a multivariate regression that models the price of Bitcoin with scarcity, the difficulty of mining, and investor attention. Further, this paper aims to show that Ridge Regression is preferred over OLS for predicting the price of Bitcoin.

1.2 Background

Bitcoin was written by an anonymous individual named Satoshi Nakamoto; likewise, one of the most promising Bitcoin pricing models comes from anonymous author PlanB's article "Modeling Bitcoin Value with Scarcity" (2019). The Stock to Flow model aims to model the market capitalization of Bitcoin by using a metric for scarcity - the stock-to-flow ratio. As previously stated, the stock to flow ratio has been used to model the value of other scarce assets, such as precious metals, in (Adrangi et al., 2015). Yet, the stock-to-flow ratio is unique in the cryptocurrency market, as it is set to be programmatically increase roughly every four years based on the Bitcoin code - not the action of the market participants. No single entity can change this, which makes it of particular interest. Within the scope of this paper, the stock-to-flow ratio is a constructed from on-chain data. Thus, the stock-to-flow ratio was chosen as a predictor variable because of the promising results that other authors have found in analyzing this metric for provably scarce assets.

We will explore another on-chain metric: the difficulty of mining. The process of mining Bitcoin involves solving a computationally difficult algorithm (SHA-256d). That is to say - the inputs of production are electricity and robust computer hardware. The mechanics of mining Bitcoin are such that when there are more miners on the network,

the mining difficulty rises, which makes mining Bitcoin more difficult for all participants (more hashes need to be submitted to find a block). Generally, the mining difficulty should rise when the market price of Bitcoin is relatively high (it is profitable to mine Bitcoin), and the mining difficulty should fall when the price is relatively low (it is unprofitable to mine Bitcoin). Notably, miners represent a class of investors who have a larger stake and financial risk tied to the Bitcoin network, as they have recurring electricity costs and physical assets to maintain as well as the price risk of the underlying asset, rather than just the price risk of the underlying for the average non-mining Bitcoin investor. As such, the inflow and outflow of miners as measured by mining difficulty should be useful in predicting the direction of prices, as inefficient miners exiting the market are likely to sell their stock of Bitcoin to cover their costs. Therefore, based on the economic intuition outlined above, mining difficulty is included in the model in order to further research on its predictive power.

Liu and Tsyvinski (2018) found that traditional asset pricing models such as CAPM fail to comprehensively and statistically significantly model the price or market cap of Bitcoin; rather, they found that momentum and investor attention were statistically significant predictor variables. As such, the model includes the worldwide Google Trends score for the search term "Bitcoin".

There are some important assumptions and conditions which needed to be addressed before the model was built. Primarily, the Google Trends data is taken over each block halving period, equal to 210,000 blocks. Google Trends data is fit to each time frame that is selected, such that the peak in interest on that time frame is always 100. This is where the term "stochastic" in the model name comes from - investor attention is random and measured from 0 to 100. By segmenting the Google Trends data into each halving period, the groups of investors who were interested in mining Bitcoin during each halving period are separated, as each group of investors (and people searching for the term) would have a distinct appetite for risk. The relationship with Google Trends data is also stronger when

broken up this way. Overall, our model assumes that the price of Bitcoin is driven by three distinct factors: (1) the scarcity of the asset, (2) the relative difficulty of producing the asset, and (3) the attention that investors give to the asset. The stock-to-flow ratio and difficulty of mining are both endogenous to Bitcoin; investor attention is exogenous.

2 OLS Regression

After building the intuition for the predictor variables in our model, we ran a simple OLS regression to see if we were on the right track. We performed a log transformation on the market capitalization, stock-to-flow ratio, and difficulty of mining. The log transformation follows the intuition from PlanB's "Stock to Flow" model and is common in Economics, especially when dealing with large numbers. The results of the multivariate OLS regression can be seen below in Figure 1:

Panel A: OLS Regression Output				
	β	SE	t	VIF
Coefficients:				
ln(S2F)	2.0661***	0.1014	17.5290	62.0147
ln(diff)	0.2314***	0.0140	18.3429	68.5799
Trend	0.0324***	0.0017	21.9113	2.2377
Intercept	11.283	0.1114	102.5654	
Model Summary:				
R^2	0.9745			
Adjusted R^2	0.9743			
Standard Error	0.5763			
RMSE	0.7073			
F-Stat	5871.712			

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Figure 1: OLS Model Specifications

Based on this output, and after applying some simple algebra (note that Total Supply

was not included in the regression, but rather pulled from the dataset), we obtain equation equation (1) for the Bitcoin Market Capitalization and equation (2) for the Bitcoin Price:

$$\widehat{Bitcoin\ Market\ Cap} = S2F^{\beta_1} \times Difficulty^{\beta_2} \times e^{(\beta_3 Trend + \epsilon)} \tag{1}$$

$$\widehat{Bitcoin\ Price} = \frac{S2F^{\beta_1} \times Difficulty^{\beta_2} \times e^{(\beta_3 Trend + \epsilon)}}{Total\ Supply} \tag{2}$$

Using equation (2), we plot the actual price and the OLS predicted price in Figure 2:

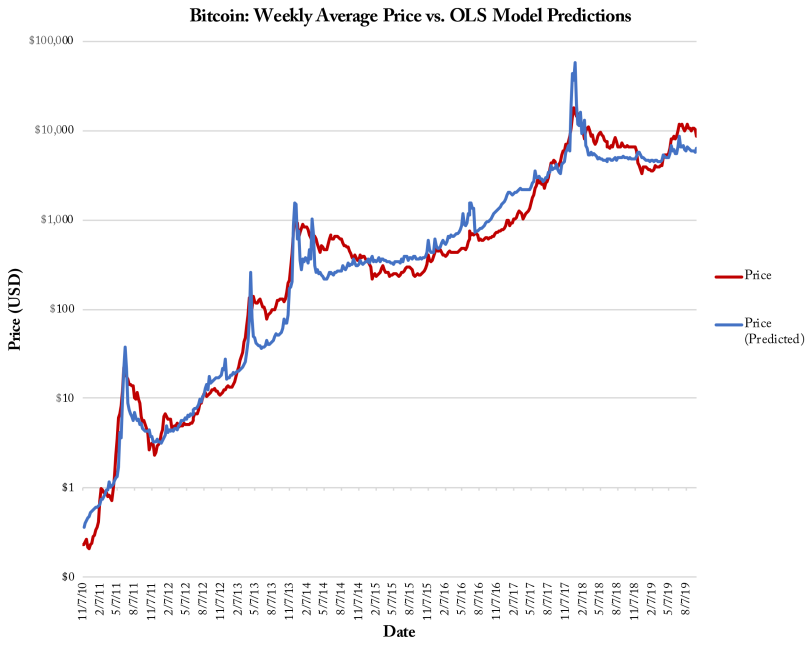


Figure 2: OLS Predictions

The plot of residual versus predicted values for the OLS model is shown below in Figure 3. Note that the residual plot is centered around zero, indicating that there is no heteroskedasticity in the initial model. The root mean squared error of the model can still be improved upon, however, and the model consistently over-estimates peaks in the price. Additionally, from Table 1, the high VIF values for ln(S2F) and ln(Difficulty) indicates that there is likely multicollinearity present in the data. As discussed below, Ridge Regression addresses both of these issues via a process called regularization. Based

on this information and the statistically significant predictor variables identified in the OLS regression, we may begin to develop the Ridge Regression.

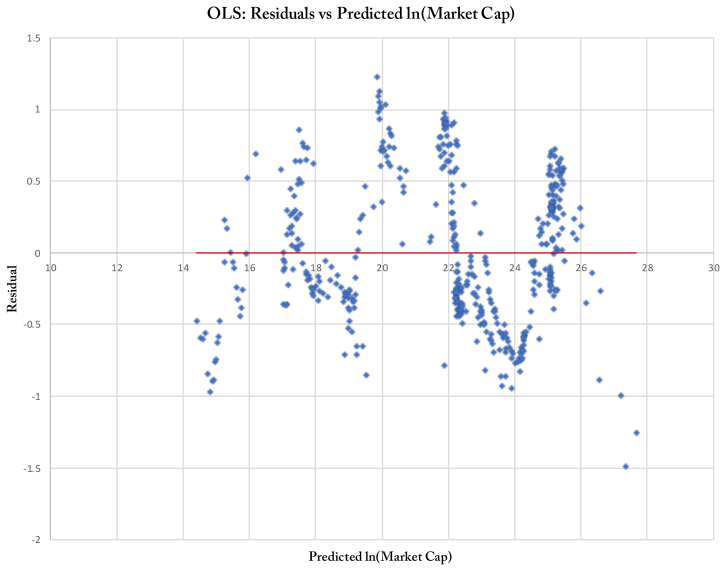


Figure 3: OLS Residuals

3 Ridge Regression

After running the initial OLS regression to ensure the intuition behind our predictor variable selection was valid, and that our predictor variables were statistically significant, we built our ML model using Ridge Regression. Ridge Regression uses a penalizing term, λ, to reduce the variance of model predictions in exchange for slightly biased estimates (the amount of bias depends on the value of λ). Additionally, Ridge helps remediate the issue of multicollinearity in linear regression (Hoerl and Kennard, 1970). Ridge Regression does this by solving the following matrix equation for estimating the beta coefficients:

$$\hat{\beta} = (X^T X + \lambda \mathrm{I})^{-1} X^T y$$

This bias-variance tradeoff is what we seek to optimize in our research in order to improve the model predictions. First, the data was split into a training set and a testing set. Given the above equation, the value of the beta coefficients are a function of λ, which is unknown. As such, a Python package called GridSearchCV was used to perform k-fold cross validation on a grid of 100 potential λ values in order to find the value of λ that yields the best predictions, as measured by mean squared error. Each of the 100 λ's was then tested on each of the 10 folds, for a total of 1,000 models being tested. The model that yields the best predictions was then selected, as measured by mean squared error (there are a number of scores that the model could have been optimized for including R-squared, but mean squared error and root mean squared error was chosen because it matters most for predictions). This cross-validation process is similar to bootstrapping. Additionally, before the multivariate Ridge Regression was performed, a Ridge Regression was performed on each of the predictor variables individually, the results of which can be found in Appendix A.

Because the end goal is to produce the lowest mean squared error model (and the because beta estimates are already biased by the λ term), additional complexity was added to the model by creating polynomials of the third degree on all of our predictor variables. This is a common transformation for Ridge Regression practitioners to perform, as the computational difficulty in solving for so many variables is trivial for a modern computer. This process usually leads to better predictions as measured by mean squared error - especially in the instance of variables that are as volatile as the Bitcoin market capitalization. Ridge Regression helps address the issue of multicollinearity via the process of penalizing the beta estimates for better predictions, unlike OLS. That is to say - the beta estimates for variables with multicollinearity will be penalized by λ such that the effect of the multicollinearity on the model predictions is diminished.

We then optimize the Ridge Regression for the best λ on this set of polynomial predictor variables. Notably, the best λ is 0.0001, which means that the beta estimates are not biased

greatly. The final Ridge Regression leads to the following output of predicted betas in Figure 4:

Panel A: Ridge Regression Output

1st degree polynomials	β	*p-value*	*2nd degree polynomials*	β	*p-value*	*3rd degree polynomials*	β	*p-value*
Intercept	9.3126		$\ln(S2F)^2$	0.3057	0.8990	$\ln(S2F)^2 * \ln(Difficulty)$	0.002239	0.9928
ln(S2F)	2.3403	0.5936	$\ln(Difficulty)^2$	-0.0122	0.00086***	$\ln(S2F)^2 * Trend$	0.010621	0.7119
ln(Difficulty)	0.0122	0.000004***	$Trend^2$	-0.0009	0.000066***	$\ln(Difficulty)^2 * Trend$	-0.000246	1.9984E-15***
Trend	-0.0175	0.7661	ln(S2F)*Trend	-0.0079	0.8681	$Trend^2 * \ln(S2F)$	0.000002	0.9837
			ln(S2F)*ln(Difficulty)	-0.1065	0.6377	$Trend^2 * \ln(Difficulty)$	0.000005	0.4811
			ln(Difficulty)*Trend	0.5084	0.1507	$\ln(Difficulty)^2 * \ln(S2F)$	0.0025	0.5769
						ln(S2F)*ln(Difficulty)*Trend	-0.0016	0.5091
						$\ln(S2F)^3$	0.0074	0.9970
Model Summary:								
R^2	0.9836					$\ln(Difficulty)^3$	0.00026	0***
RMSE	0.4023							
Best λ	0.0001					$Trend^3$	2.178E-06	2.699E-09***

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Figure 4: Ridge Regression Output

Interpreting the output from Figure 4 leads us to the functional form for the predicted natural log of the Bitcoin market capitalization in equation (3), below:

$$\begin{aligned}
\ln(BTC\ \widehat{Market\ Cap}) = {} & \beta_1 \ln(S2F)^3 + \beta_2 \ln(Difficulty)^3 + \beta_3 Trend^3 \\
& + \beta_4 \ln(S2F)^2 \times \ln(Difficulty) + \beta_5 \ln(S2F)^2 \times Trend \\
& + \beta_6 \ln(Difficulty)^2 \times \ln(S2F) + \beta_7 \ln(Difficulty)^2 \times Trend \\
& + \beta_8 Trend^2 \times \ln(S2F) + \beta_9 Trend^2 \times \ln(Difficulty) \\
& + \beta_{10} \ln(S2F) \times \ln(Difficulty) \times Trend + \beta_{11} \ln(S2F)^2 \\
& + \beta_{12} \ln(Difficulty^2 + \beta_{13} Trend^2 + \beta_{14} \ln(S2F) \times \ln(Difficulty) \\
& + \beta_{15} \ln(S2F) \times Trend + \beta_{16} \ln(Difficulty) \times Trend + \beta_{17} \ln(S2F) \\
& + \beta_{18} \ln(Difficulty) + \beta_{19} Trend + \epsilon
\end{aligned}$$

This functional form was kept for the sake of simplicity, and the values of the natural log of the predicted market capitalization were stored in a dataframe for further interpolation. Some simple algebra was then performed to come up with the predicted Bitcoin price equation found in equation (3) below:

$$\widehat{Bitcoin\ Price} = \frac{e^{\ln(\widehat{Ridge\ BTC\ Market\ Cap})}}{Total\ Supply} \tag{3}$$

These results are quite promising, and show that the intuition in creating the polynomial grid earlier was correct. Notably, the R-squared value and the root mean squared error of the Ridge model have improved over the base OLS model, which has a root mean squared error of 0.7591. We take note of the fact that, by adding in an arbitrary amount of complexity to the model with the polynomial terms of degree n, the root mean squared error of the model could be further diminished, which is in line with the assumptions made. Therefore, the performance of the model is a function of both lambda and the polynomial degree n. However, it was determined that the functional form of a 10-degree polynomial, for example, is perhaps a bit burdensome and difficult to build intuition from, despite the predictive power. The relationship between polynomial degree and root mean squared error can be seen below in Figure 5, below:

Panel A: Ridge Regression Polynomial Degrees				
Polynomial Degree	*1*	*3*	*5*	*10*
Ridge Parameters:				
λ	0.0514	0.0001	0.0001	0.0001
R^2	0.9768	0.9836	0.9891	0.9929
RMSE	0.5642	0.4023	0.3495	0.3131
Intercept	11.3298	9.8447	6.9503	8.0548

Figure 5: RMSE decreases as polynomial degree increases

The weekly average Bitcoin price versus the Ridge Model predicted price is then plotted. Visually interpreting the output in Figure 6, one can see that the Ridge Regression model is indeed better than the OLS model at predicting the weekly average price of Bitcoin. Notably, the Ridge model performs significantly better than the OLS model at peaks in the price, which was as issue that was identified in the OLS. This issue was likely caused by the multicollinearity that was identified, and shows the efficacy of Ridge in dealing with problems like this.

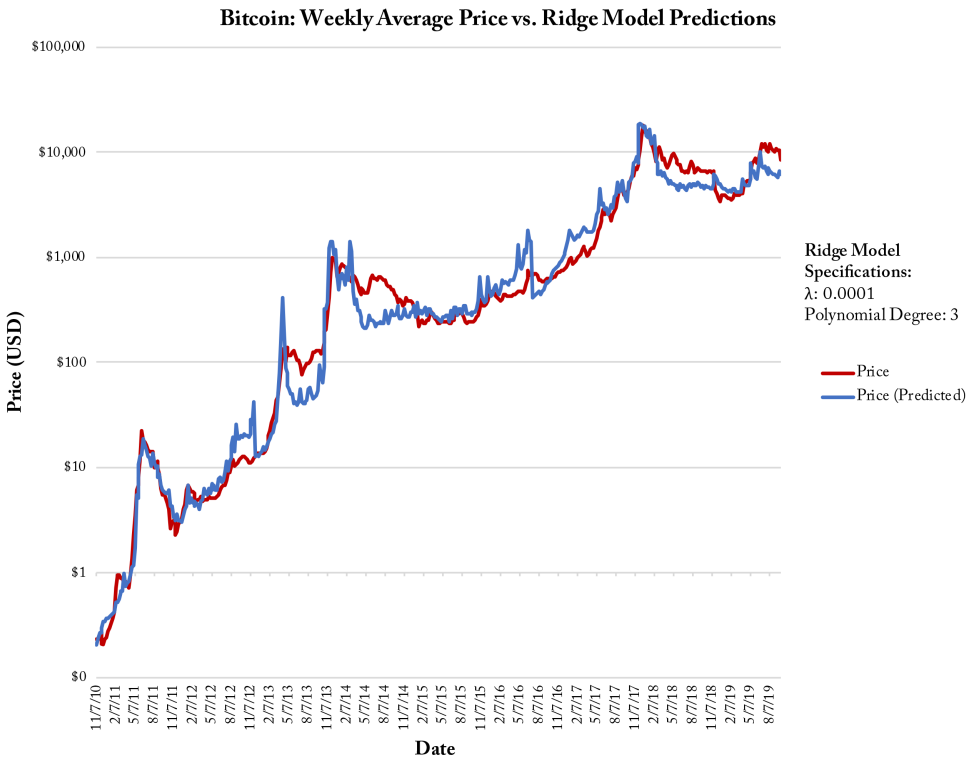


Figure 6: Ridge Regression Predictions

Checking the residual versus predicted values for the Ridge Regression yields similar results to the OLS regression. No hetereoskedasticity is present in the residuals, plotted in Figure 7.

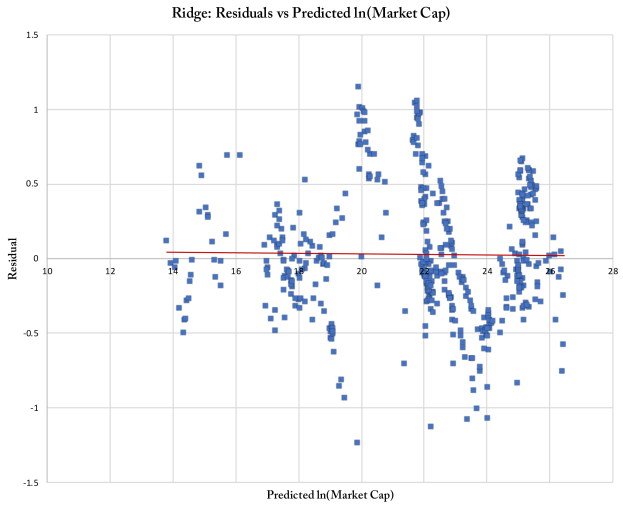


Figure 7: Ridge Regression Residuals

3.1 Out of Sample Performance

The training data was purposefully cut off on October 1, 2019 to allow for adequate time for performance testing. Plotted below in Figure 8 are the out-of-sample model predictions for the Ridge Regression model and the OLS regression model. Visually interpreting the output, it can again be seen that Ridge Regression is superior to OLS regression in predicting the weekly average price of Bitcoin. Output from other models that were studied is included for reference in Appendix B.

Notably, the stock-to-flow ratio changes discontinuously on the date of the Bitcoin "halving" - when the new supply issuance of Bitcoin is cut in half, roughly every four years. This causes a momentary degradation in model performance. The most recent halving happened to occur on May 11, 2020, and as such has not been included in this set of out-of-sample predictions. Once more time has elapsed, and data after the Bitcoin halving can be included in the training set, the Ridge model should be used to predict on this new data. This was simply outside of the time constraints imposed upon the research.

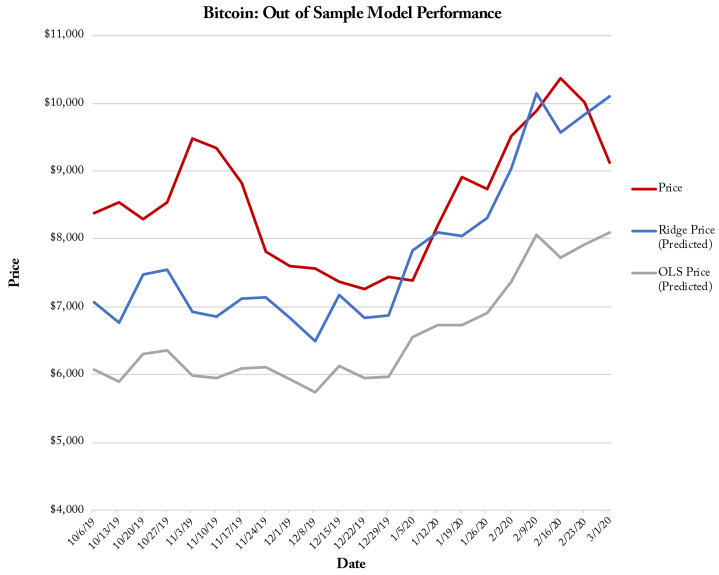


Figure 8: Ridge and OLS Out of Sample Predictions

4 Conclusion

In this paper, the weekly average price of Bitcoin was modeled using three distinct market-specific factors: the stock-to-flow ratio, mining difficulty, and investor attention. Ridge Regression was used to build a low root mean squared error predictions model, which was shown to be superior in its predictive power compared to that of an OLS regression using the same predictor variables. The stock-to-flow ratio and the mining difficulty were found to have a high VIF, indicating multicollinearity, which provides further justification for using Ridge Regression to predict the price of Bitcoin. It was shown that Ridge Regression benefits from additional degrees of complexity created from a polynomial grid of the predictor variables. These results further establish the fact that while Bitcoin may draw comparisons to other asset classes (such as precious metals) it is completely unique in its own right and requires its own distinct analysis.

A consideration that should be made about this model is how useful it is in forecasting, as opposed to making week-to-week predictions. We have displayed the out-of-sample performance of the model, which shows promise in making week-to-week predictions. However, only the stock-to-flow ratio can be reliably forecasted due to its programmatic

nature in the Bitcoin code. The mining difficulty is dynamically adjusted every 2016 blocks (a Bitcoin block is mined, on average, every 10 minutes), per Nakamoto (2008). Google Trends data is published every week and cannot be reliably projected. As such, the Ridge Regression model is unsuited for making long-term forecasts about the Bitcoin price. This does not diminish the predictive power of the Ridge model on a weekly basis, however. A practitioner could simply predict on new data when it is published.

A common criticism of ML models such as Ridge is that they rely heavily on re-sampling data in order to make predictions, which makes them not ideal for model classification. It could be the case that without re-sampling, over time the performance of the Ridge model degrades. For practitioners, this is an easy problem to fix. A practitioner can merely re-sample the data if the predictions are not satisfactory, if their sole aim is to consistently produce the best predictions. This "fix" does make it difficult to state that any one combination of beta estimates is the"best", however.

Unfortunately we were unable to gain access to Twitter API data within the specified time frame of this research, due to questions and concerns from Twitter over who would be accessing the data. The reason for this is outside the scope of this paper, but it should be noted that Twitter API data is no longer as easy to access, and as such, only Google Trends data represented investor attention in the model. Ideally, Twitter tweet volume data would have complemented this metric. Because of this, Twitter data warrants further research with Ridge Regression.

Overall, we achieved our objective of building a low root mean squared error predictions model for the weekly average price of Bitcoin. The model proves that mining difficulty is a statistically significant predictor of the Bitcoin price, which provides further opportunities for research. The root mean squared error of the Ridge Regression is 63% smaller than that of the root mean squared error of the OLS model, which demonstrates the robustness of the Ridge model and the benefit of adding additional complexity to an ML model for the sake of predictions. CAPM and Fama French models from Liu and

Tsyvinski (2018) only reported R-squared values, with a maximum of 0.05 for the 5 and 6-factor models. The Ridge model does significantly better, with an R-squared value of 0.9836. However, it should be noted that by definition, the beta estimates of the Ridge Regression are biased (for the sake of reduced variance). Ridge Regression is one of many machine learning methods available to researchers, and research on additional cryptocurrency market-specific factors should be performed using a variety of machine learning techniques, such as Lasso for feature selection. This paper has demonstrated the promise of machine learning models when applied to the Bitcoin market.

References

Abraham, J., Higdon, D., Nelson, J., and Ibarra, J. (2018). Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1.

Adrangi, B., Chatrath, A., Christie-David, R. A., Miao, H., and Ramchander, S. (2015). Stock-versus-flow distinctions, information, and the role of inventory. *Journal of Futures Markets*, 35(11):1003–1025.

Bernanke, B. S., Boivin, J., and Eliasz, P. (2005). Measuring the effects of monetary policy: a factor-augmented vector autoregressive (favar) approach. *The Quarterly journal of economics*, 120(1):387–422.

Goczek, Ł. and Skliarov, I. (2019). What drives the bitcoin price? a factor augmented error correction mechanism investigation. *Applied Economics*, 51(59):6393–6410.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Liu, Y. and Tsyvinski, A. (2018). Risks and returns of cryptocurrency. Technical report, National Bureau of Economic Research.

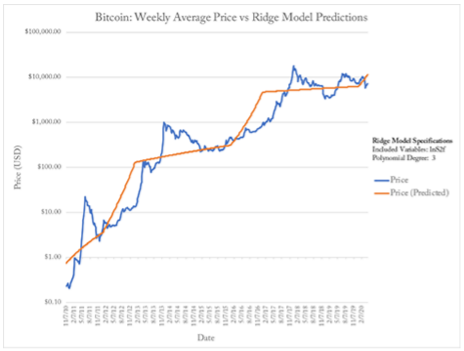
Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., and Mozetič, I. (2015). The effects of twitter sentiment on stock price returns. *PloS one*, 10(9).

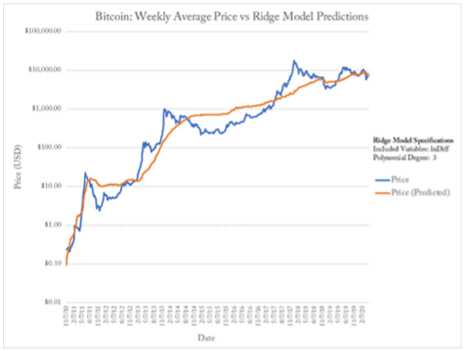
Taylor, M. B. (2013). Bitcoin and the age of bespoke silicon. In *2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, pages 1–10. IEEE.

A Ridge Model Testing

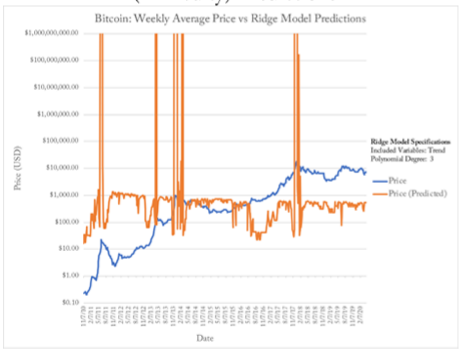
(i) Model Testing: Predictoins



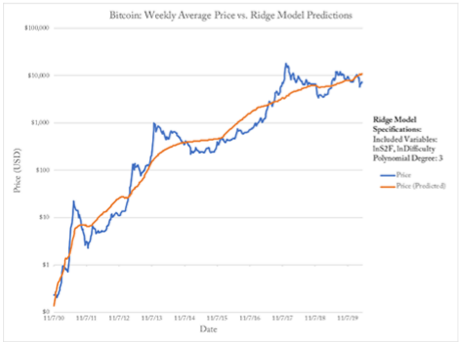
ln(S2F) Predictions



ln(Difficulty) Predictions

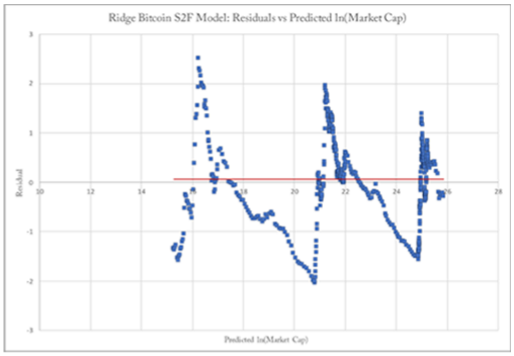


Google Trends Predictions

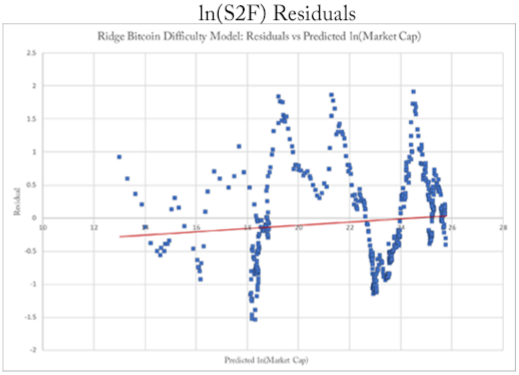


ln(S2F) and ln(Difficulty) Predictions

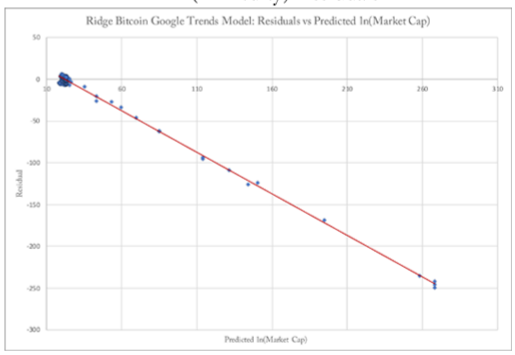
(ii) Model Testing: Residuals



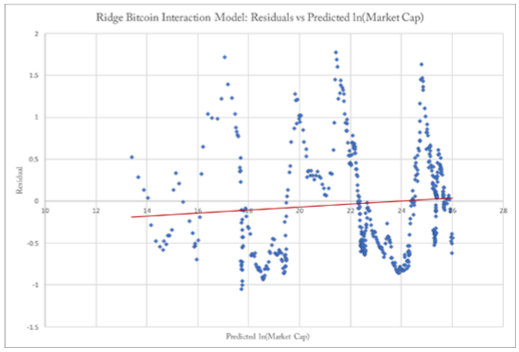
ln(S2F) Residuals



ln(Difficulty) Residuals



Google Trends Residuals



ln(S2F) and ln(Difficulty) Residuals

B Other Models

Table 5: Bitcoin Returns Factor Loadings

(Percentage)	CAPM	3-Fac	4-Fac	5-Fac	6-Fac
ALPHA	18.91**	18.20**	17.66**	16.72**	15.98*
	(2.42)	(2.30)	(2.18)	(2.07)	(1.94)
	[2.55]	[2.34]	[2.28]	[2.61]	[2.54]
MKTRF	3.34	3.79	4.00	4.57*	4.85*
	(1.45)	(1.56)	(1.60)	(1.81)	(1.86)
	[1.94]	[2.08]	[1.94]	[2.14]	[2.06]
SMB		−1.29	−1.26	0.45	0.55
		(-0.36)	(-0.35)	(0.12)	(0.14)
		[-0.55]	[-0.54]	[0.15]	[0.16]
HML		−3.02	−2.41	−3.80	−3.01
		(-0.81)	(-0.59)	(-0.79)	(-0.58)
		[-1.22]	[-0.84]	[-0.97]	[-0.67]
MOM			1.08		1.35
			(0.38)		(0.47)
			[0.48]		[0.59]
RMW				6.16	6.39
				(1.07)	(1.10)
				[1.35]	[1.41]
CMA				2.47	2.40
				(0.35)	(0.33)
				[0.27]	[0.24]
R-Squared	0.02	0.04	0.04	0.05	0.05

Figure 9: CAPM Results from Liu and Tsyvinski (2018)

Table 20: Google Searches by Groups

Weekly Level (Percentage)							
Rank	Google	R_{t+1}	T-Statistics	Sharpe	R_{t+2}	T-Statistics	Sharpe
Low	−0.71	1.07	(0.74)	0.08	0.34	(0.23)	0.03
2	−0.05	−1.20	(-1.06)	-0.11	0.24	(0.20)	0.02
3	−0.01	3.92**	(2.26)	0.24	4.23***	(2.75)	0.29
4	0.04	6.03**	(2.65)	0.35	5.21**	(2.36)	0.31
5	0.87	11.20***	(3.95)	0.48	8.99***	(3.17)	0.39
Difference		10.13			8.66		
Rank	Google	R_{t+3}	T-Statistics	Sharpe	R_{t+4}	T-Statistics	Sharpe
Low	−0.71	−0.29	(-0.19)	-0.02	0.62	(0.39)	0.05
2	−0.05	1.65	(1.47)	0.15	1.31	(1.03)	0.11
3	−0.01	4.35**	(2.39)	0.25	4.49**	(2.57)	0.27
4	0.04	6.19***	(3.06)	0.40	9.13***	(3.33)	0.44
5	0.87	6.54**	(2.39)	0.29	3.51	(1.67)	0.20
Difference		6.82			2.89		

Figure 10: Investor Attention Results from Liu and Tsyvinski (2018)

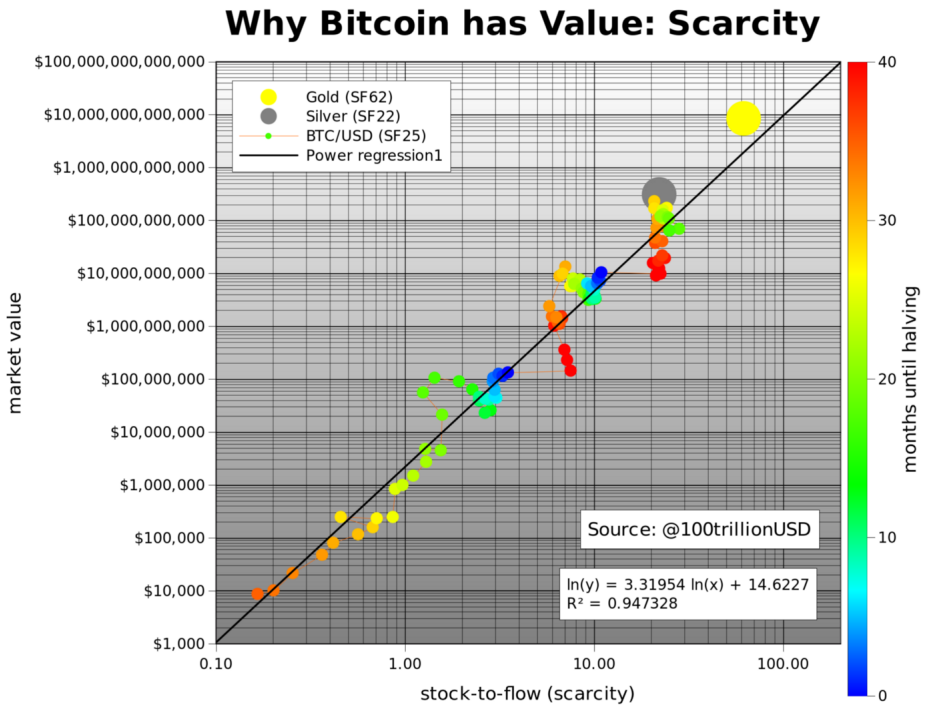


Figure 11: The Original S2F Model, from PlanB (2019)