University of Strathclyde

Department of Accounting and Finance/ Department of Management Science

M.Sc. Financial Technology (2017/2018)

"Intrinsic Value of the Ethereum Blockchain Network"

EPS Ethereum =

ethereum

Management Science Project

Joshua Eick

Student No. 201779687

Submitted in partial fulfilment of the requirements for the degree of M.Sc.

August 15, 2018

Table of Contents

Abstract

Cryptocurrency is starting to be considered as an asset class for investment portfolios because of the multiple competitive advantages it has and its beneficial correlation to other asset classes. Most investors in cryptocurrency are speculators driven by market sentiment, investing according to technical analysis. There is a gap between technical analysis and fundamental analysis in the area of cryptocurrency. With the adoption of fundamental analysis the real intrinsic value of cryptocurrency can be achieved with higher returns being gained.

This research aims to identify key variables and valuation metrics of Ethereum Blockchain Networks in order to predict the intrinsic value of ether through linear multiple regression. This will involve presenting a model including fundamental variables of the Ethereum Blockchain Network and market sentiment with the objective of achieving higher returns for investors of ether. There will be a focus on fundamental analysis, rather than technical analysis, of cryptocurrency because it is presume that has a greater relation to the intrinsic value of cryptocurrency.

Based on the research's unsupervised method of linear regression, a price prediction model of ether with a Mean Sum Square Error of 1.1266*e^-6 and R square of 99% is devised. The results indicate that the features of the Ethereum Blockchain Network and valuation metrics have more predicting power than the market sentiment (Crix-Crypto Index). The research highlight that the most significant variable to ether are gas price per block, transactions fees and reward to miners, and focused on the utility of ether which can be of intrinsic value and have a significant impact on investment portfolios.

Introduction

"Price is what you pay, value is what you get." (Warren Buffett, 2014). What is the value?

In exploring the value of ether cryptocurrency for cryptographers, technologists and investors we can start with that point that it is beginning to be considered as the new asset class for the investment portfolios. It is becoming more common that Investment Portfolio Managers, Hedge Fund Managers, ETFs, open and closed investment funds want to open new funds with cryptocurrencies because of the potential positive impact that can be delivered to the portfolio, such as correlations. This has lead to Bloomberg and Galaxy Digital Capital Management LP developing an index for cryptocurrencies (Bloomberg.com, 2018). On the other side of the coin, De (2018) highlights that William Hinman, Securities Exchange Commission (SEC) Director of Corporate Finance make an statement that ether cannot be considered as securities transactions. The Securities Exchange Commission, therefore, does not approve of ether as a security in United States. However, it seems likely that blockchain technology will be embraced, supported and approved by the government because the positive impact it is having on the economy. The blockchain technology that is behind the cryptocurrency is considered to be the new backbone of financial transactions that is disrupting the financial industry. This blockchain technology creates a new market, which is a new unique asset class for investment portfolios. With these facts, broadly speaking investors are asking if Ethereum cryptocurrency can behave as security and what its intrinsic value is which can help achieve higher returns.

There is a big gap between technical analysis and fundamental analysis in the area of cryptocurrencies. A lot of technical analysis has been done because the investors in cryptocurrencies are generally speculators. In contrast, if we turn to looking at cryptocurrency as an asset class for investment portfolios, there is a need for fundamental analysis. Because most of the academic research that has been done is technical/market analysis, there is fundamental valuation gap of valuations and ratios. Investors have to understand that each cryptocurrency has a utility, which is the intrinsic value. It would be beneficial if a greater amount of research was done that examines the real intrinsic value of individual cryptocurrencies.

The majority of academic research and content about price predictions in cryptocurrency has focused on Bitcoin. However, there is limited research about Ethereum, which is the second biggest market cap of cryptocurrency. Ether, the coin of Ethereum, delivered another value in the Blockchain Network, which is the feature of functionality. The Ethereum functionality feature allows people to develop decentralized applications (Dapps) in the Network that generate business for the economy. This functionality feature of ether is transforming how people are doing business. Therefore, it is important to calculate the value that the functionality feature adds to the Blockchain Ethereum Network. This is done by through fundamental analysis that helps to calculate the intrinsic value of ether. This research has selected ether as the basis for a development model that predicts the price of ether because of his functionally feature and its ability to disrupt business.

This research is written for the consideration of portfolio investment managers and investors. Currently, people in Wall Street wait for the quarterly reports of the equity companies, which contain financial information like revenues, cost, earnings and EPS. Because it is totally new cryptocurrencies space have very limited quarterly reports for investors. This research, therefore, aims to identify the key/significant variables and develop valuation ratios correlated to the price of ether in order to predict prices and achieve higher returns. In other words, the study will develop valuations metrics and ratios that represent the properties of the Network (scalability, decentralization, security) and attributes of Blockchain (cost, speed, transparency) that have the power to predict the price of ether. An example of this is ratios of cost like gas price per transaction. In addition, identifying key variables to make ratios like hash rates, is extremely significant to the network. The 51% of hack attacks by miners highlights how significant features of the network can be to its price. In addition, adopting the network value to transactions ratio (NTV) to the model, which is the equivalent of the PE ratio in equity, as developed by Woo (2017), would appear to be significant to this model. Another key variable to consider is market sentiment, which is widely agreed to be one of the most significant variables influencing the prices of cryptocurrencies. Moreover, the results of research by Sorgente and Cibils (2014), outline that the blocksize is a key indicator to the price of the Bitcoin Blockchain Network. This research will incorporate in the model all these significant variables that are key to increasing the prediction power of the price of ether.

Figure 2.1- Ether price through time

An interest in incorporating cryptocurrency as an asset class and calculating the intrinsic value is clear and also key for investors looking to achieve higher returns and re-asset allocation for investment portfolios.

Figure 2.1 illustrates why investors want to ride this bull market. There are substantial wave price increases of ether prices through time, which achieve high returns. The company Invest in Heaven, for instance, is the first in the world to offer a blockchain investing research service that estimates that the price of ether I is going to be around $1,000 between 2018 and 2020 (Investing Haven, 2017). In comparison, Bloomberg news highlights that Tom Lee, one of the most well-known and highest ranked institutional investors in Wall Street, predicts that ether price by the end of this year (2018) will be $1,900 (Bloomberg, 2018). There are then, different predictions about the price of ether, with some conservative and others extremely aggressive.

The most significant factor in predicting is what model to select that incorporates all the factors affecting price. Essentially the model will impact on the investment decision made by investors. In fact, Bloomberg (2017) decides that one of the best models for cryptocurrency is the four factors model, which considers size, quality, service and coins. In contrast, Investing Heaven (2017) stipulates the three factors model considering demand (transactions), supply and number of applications developed on top of Ethereum. Other investment companies incorporate other models. This research considers supervised methods like linear regression because of the competitive advantage they have for the objectives of this research, which is to explain what the valuation variables are that have more predicting power to price that can be fitted to a model.

Overall, to identify significant variables and develop valuation metrics of the properties of the network and attributes of the Blockchain, take into account market sentiment variables like cryptocurrency index (the value of the top 20 market cap cryptocurrencies like Crix Crypto-Index developed by Trimborn and Härdle , 2016) and incorporating them into a new model that can achieve higher returns. It's believed, because of the simplicity of the approach, this will contribute and add extra value to fintech companies (robot-advisor) and the investment industry in the new asset class of cryptocurrencies, particularly in relation to the Proof of Work mechanism. At the end of the day, investors can use this model to maximize their returns, with the intention of this research being to change the mindset of how investors analyze cryptocurrency through the approach of fundamental analysis.

Literature Review

Previous academic research related to this investigation is limited because cryptocurrency is a new technology, about which information is hard to obtain. Most related research on cryptocurrency is based on the Bitcoin cryptocurrency because this was the first cryptocurrency in the area. However, there is limited research concerning Ethereum Blockchain Network, and the focus of the research that does exist focuses on technical analysis (market sentiment) rather than fundamental analysis. This suggests that there is a lot of room to explore with Ethereum Blockchain Network. In fact, the major authors in the academic research of cryptocurrencies related to supervisory methods are from the University of Stanford and MIT. It is worthwhile giving a brief account of these to help provide greater context for the current study.

In one of the first studies in the area, that of Shah and Zhang (2014), the authors tried to predict the price of cryptocurrency using supervised methods of machine learning techniques with the Bayesian model and binary classification. The study argued that this model could double returns in around 60 days with Bitcoin. A strategy was created to predict the return of Bitcoin and found that the Bayesian model was suitable for achieving returns in cryptocurrency. However, the study did not examine other models in order to compare their accuracy and potential for considering prediction prices.

A key piece of research is that by Greaves and Au (2015), which outlined the use of supervised methods like linear regression, and how they outperformed other regressions models with MSE of 1.94, in predicting the price of Bitcoin with the features of the network. In addition, the authors identified the highest classification model with accuracy of 55% as being through the Neural Network (Greaves and Au, 2015). The general conclusion of this research was that it is necessary to incorporate market sentiment to increase predicting power. This study mention previous research by pointing out that the Exchange data of cryptocurrencies has strong predicting power, (Greaves and Au, 2015). This research backed up the premise of the current study, that selecting linear regression is a superior approach for predicting price in cryptocurrencies. It also supported the view that choosing fundamental values from the network and market sentiment are key for predicting power. Certainly, adding fundamental ratios of the property of the Ethereum Network will add accuracy to the model. Thus, the investors and portfolio managers using this model can achieve higher returns.

Another key academic study, this time by Madan, Saluja and Zhao (2014), adds extra value to the research by Greaves and Au (2015) mentioned above and increases the accuracy level with the supervised method of classification up to 97% with binomial logistic regression trying to predict Bitcoin price. This research suggests that the key difference to adding greater accuracy in predicting the sign of daily price changes, was to take into account 16 features of the network. This research highlights that one of the vital factors in gaining more predicting power is the large features of the network. This is also relevant to the current paper, which also takes into account features of the network and ratios with the features of the network. However, Madan, Saluja and Zhao question the model, which chooses linear regression over binomial logistic regression, which they argue achieves higher accuracy. Thus, it's found that supervised methods performed well but the question is what methods is better off.

The prior research in the area, which is most significant for the present study, is that which predicts the price of Ethereum instead of Bitcoin. Chen, Narwal and Schultz (2017), do this, but focusing only on the variable of prices and comparing them with other models (the best performance achieved with ARIMA had 61% of accuracy).
This indicates that the outcome of only taking the variable of price movement into account to predict Ethereum is not completely accurate, because the percentage of relatively low. This adds weight to the basis of the present study, which is that to predict the price of Ethereum it is a requirement to include fundamental features of the Network. However, this research highlighted that the ARIMA model outperformed other regression models (as illustrated in table 2.1). In contrast, according to research by Amjad and Shah (2017), classification algorithms and learning the empirical conditional distribution outperformed ARIMA for predicting Bitcoin prices because of the limitations. Therefore, because of the limitations of the ARIMA model, it seems that it is not superior to other models. The limitations in this case are the assumptions that are met when applying ARIMA model. From these two pieces of research it is deduced that prices of ether as the only factor cannot predict well the prices of cryptocurrencies. Furthermore, taking classification models instead of real value models to predict price is problematic with only the variable of prices.

Table 2.1 – Accuracy of the research model Predicting Price Changes in Ethereum (Chen, Narwal and Schultz, 2017)

Method	Accuracy
Logistic Regression	53.40 %
Logistic Regression [Binary]	56.94 %
Naive Bayes	51.78 %
Support Vector Machines	51.29 %
Support Vector Machines [Change]	52.59 %
Support Vector Machines [Binary]	55.99 %
Random Forest	50.81 %
ARIMA	61.17 %
Recurrent Neural Network	52.43 %
Neural Network	52.18 %

Figure 4: Ether Price Change Predictor Accuracies

Another significant variable to consider is market sentiment in the model because in the area of cryptocurrency, it is widely agreed that the "hype" of cryptocurrency is strongly correlated to price. Research by Lamon, Nielsen and Redondo (2017) outlined that the predicted price fluctuations for Ethereum using different classification models. The variables used were news and social media data, which were accurate in estimation price increases however the estimation of price decreases where not that accurate with the Bayes method (Nielsen and Redondo, 2017). It is inferred then, that market sentiment has to be considered in the model to increase the predicting power. It can be argued that the approach of the classification models in general are not superior in predicting prices, because they do not surpass 90% in accuracy.

Lastly, recent research by Guo and Antulov-Fantulin (2018) compares the most up-to-date models to predict Bitcoin with the fluctuations of the dollar. The results identified the best lower Mean Squared Error as the XGT model and regularized regression ENET.

Therefore, there are examples of research in the area that examine different models and the different levels of accuracy they achieve.

To conclude this literature review, it can be summarized that academic research related to cryptocurrency points to selecting high amounts of key variables of the features of the Blockchain network, market sentiment variable and models in the supervised method techniques, like regressions of machine learning techniques, as the best ways increase predicting power. It is suggested that the main factors/variables to increase predicting power highly depends on what the specific features of the network that have been selected are. The key question is what type of regression model is most suitable. According to the results of the research by Alex Greaves, Benjamin Au (2015), the linear regression performed well with minimizing Mean Square Error, which is one of the reason that this research is utilized the linear regression to predict ether.

Finally, all the prior research concentrates on the prices and features of the network for predicting prices. Therefore, exploring other areas would add extra value to the study of cryptocurrency. Examples of these include, developing valuations metrics and ratios of the property of the network (scalability, security and decentralization) and the attributes of the Blockchain (cost, speed and security). This would help higher correlations with the price of ether and perform supervised methods with the model, like linear multiple regression, which could increase predicting power and lead to a proposed model that achieves exceptional returns for investors and portfolio managers.

Research Question

The aim of the research is to develop and identify key valuations metrics for the Ethereum Blockchain Network that are correlated to price in order to propose a model that predicts the price of ether with high accuracy for investors and portfolio managers to maximize returns.

Hypothesis

The valuation metrics of the Ethereum Blockchain Network and market sentiment can when combined predict the price of ether cryptocurrency with low Mean Squared Error and high R Squared.

Methodology

Initially, the dataset of Ethereum Blockchain Network (Kaggle, 2018) will be explored and analyzed with summary statistics and visualizations that focus on the variables related to the main properties of the network, such as scalability, security and decentralization. There will also be a focus on attributes of the blockchain such as cost, speed and security. Secondly, valuations metrics and ratios of the Ethereum Blockchain Network will be developed. The study will go on to explore the correlation of the variables with prices with the dataset transformed to log 10, and the dataset without this log 10 transformation. This is done in order to identify key variables strongly correlated to price that can achieve higher predicting power for the model. Then, the supervised method such as linear multiple regression will be utilized in order to predict the price of ether (dependent variable) with the independent variables (features of the Network, market sentiment and valuations metrics). The linear regression will also be performed with data transformed to log 10 and without transformation of the data to log 10. This will help with choosing the data that fits the model best. Then, the dataset will be selected that best fits the model, and analysis will be conducted of the strongest coefficients. This will help redefine the model with the best trade-off between the independent variable numbers and the smallest Mean Square Error. A price prediction model of ether will then be proposed, after which, an overview of the findings will be presented, and reflections of the research will be made along with conclusions about the contribution the study makes to the area of cryptocurrency in general.

Variables take into account in the analysis

The variables of the features of the Ethereum Blockchain Network are taken from the dataset called "Cryptocurrency Historical Prices" (Kaggle, 2018). The market sentiment variable selected is the Crix Crypto-Index, developed by Trimborn and Härdle (2016). In this research, market sentiment is selected using the Crix Crypto-Index developed by Trimborn and Härdle (2016) because it is designed to deliver a benchmark for cryptocurrencies from the past to the present of the cryptocurrencies market.

It is selected because it has a quality dataset and the methodology has proved to be robust in creating the index and index track values since 2014. The study includes 20 market cap cryptocurrencies, which ideally represent the overall market of cryptocurrency.

The dataset analysis is from 09/01/15 to 2/20/2018. From the datasets 13 key features of the Ethereum Blockchain Network are selected; 11 variables of valuations/ratios developed and a variable of market sentiment. As a result the study arrives at 25 variables for analysis.

The variables selected in the research of the dataset are the following:

Variables of features of the Ethereum Blockchain Network

1. eth_etherprice = price of Ethereum
2. eth_tx = number of transactions per day
3. eth_supply = number of ethers in supply
4. eth_marketcap = market cap in USD
5. eth_hashrate = hash rate in GH/s
6. eth_difficulty = difficulty level in TH
7. eth_blocks = number of blocks per day
8. eth_uncles = number of uncles per day
9. eth_blocksize = average block size in bytes
10. eth_gasprice = average gas price in Wei
11. eth_gaslimit = gas limit per day
12. eth_gasused = total gas used per day
13. eth_ethersupply = new ether supply per day

Variable of market sentiment

14. Crix Crypto Index

Variables/ratios developed from the dataset for valuation

15. Sales (Transaction fees in dollars per day excluding gas limit) = gas price in Wei*.000000002* ether price in dollars

    In practice, the formula for calculating transaction fees in dollars is the following: (transaction fees in dollars = Gas price in Wei gas limit *. 000000002* ether price in dollars). In this research, the gas limit variable is excluded from the transaction fees formula because if you excluded this variable (gas limit) increases the correlation between the price of ether and cost of the transaction fees formula (without gas limit). As a result the assumption of a linear relationship in the linear regression is strongly supported with the change in the formula.

16. Cost (Reward to miners in dollars per day)= ether price/block *#blocks*ether price in dollars

    *Assumption-for the cost variable developed from the dataset, which represents the cost of mining a block, is input a constant value of 3 ether per block. In fact, when Ethereum was launched this was 5 ether block, but was later reduced to 3 ether per block.

17. Earnings =Sales-Cost
18. EPS=Earnings/Total Supply of ether
19. PE ratio=Market Value/Total transactions per day
20. Price per Sales=Price of ether / Sales

Variables /ratios of costs developed from the dataset

21. Gas price/day in dollars per transaction =Gas price*.00000002*ether price/ Transaction per day
22. Gas price/day in dollars per block=Gas price*.00000002*ether price/# blocks

Variables/ratios of security/decentralization

23. Hash rate per TPD= Hash rate / transactions per day
24. Blocksize per TPD= Blocksize/transaction per day

Variables of Speed/Scalability developed from the dataset

25. TPS=Transactions per day / seconds per day

Analysis and Results

It is first necessary to clean up the dataset. In the clean-up of the dataset the missing values are detected (solved taking the average in most of the cases) and columns/variables with no needed in the analysis are excluded (dropping). For instance, the first months of the dataset of Ethereum (Kaggle, 2018) has too many missing values so is dropped along with the analysis from 09/01/15 to 2/20/2018. The data cleanup is explained in more detail in the code (appendix 2).

Dataset

In order to gain insight, the structure of the dataset is analyzed through different visualizations.

Figure 2.2- Distribution of the Top Features of the Ethereum Blockchain Network

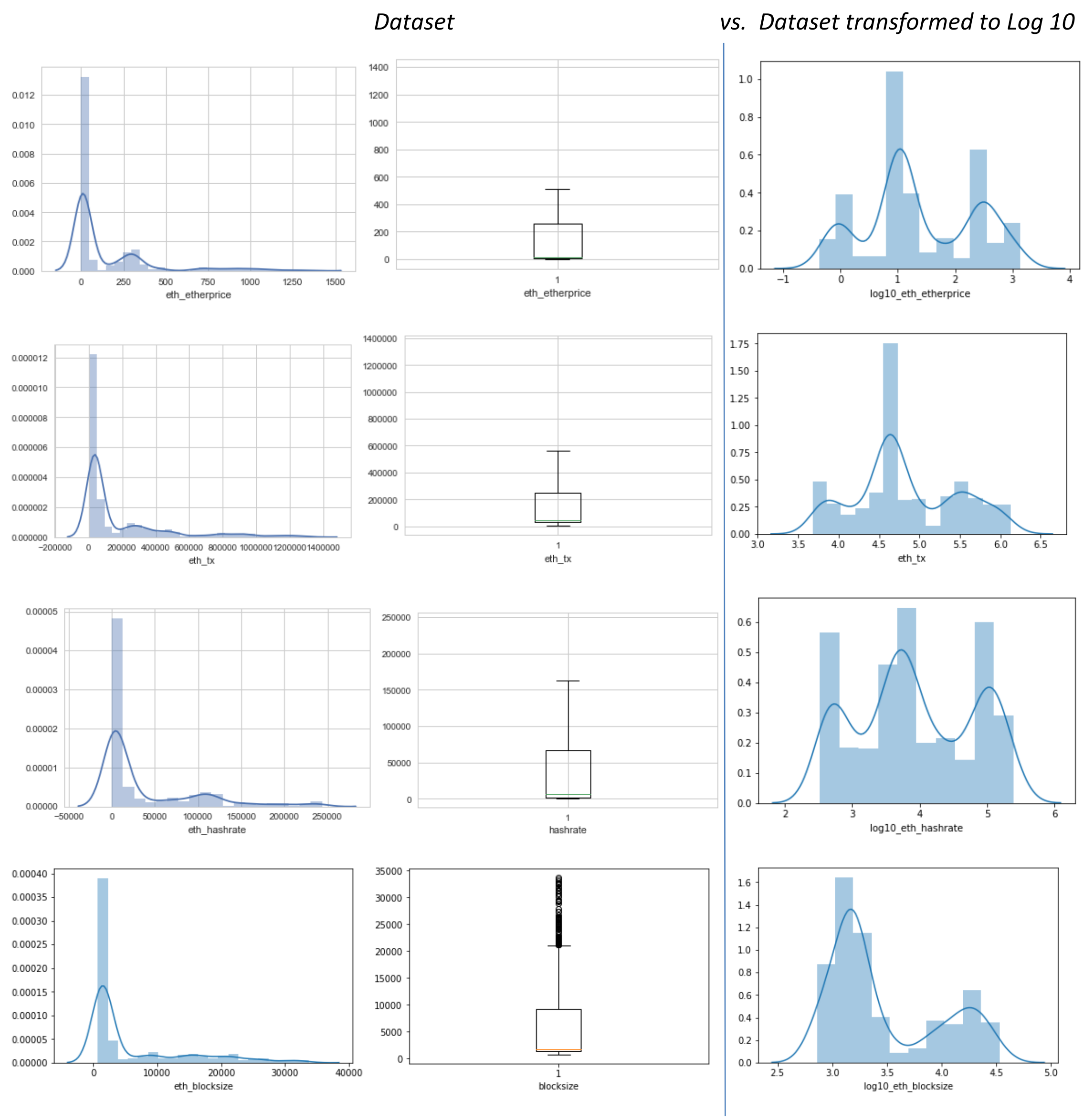
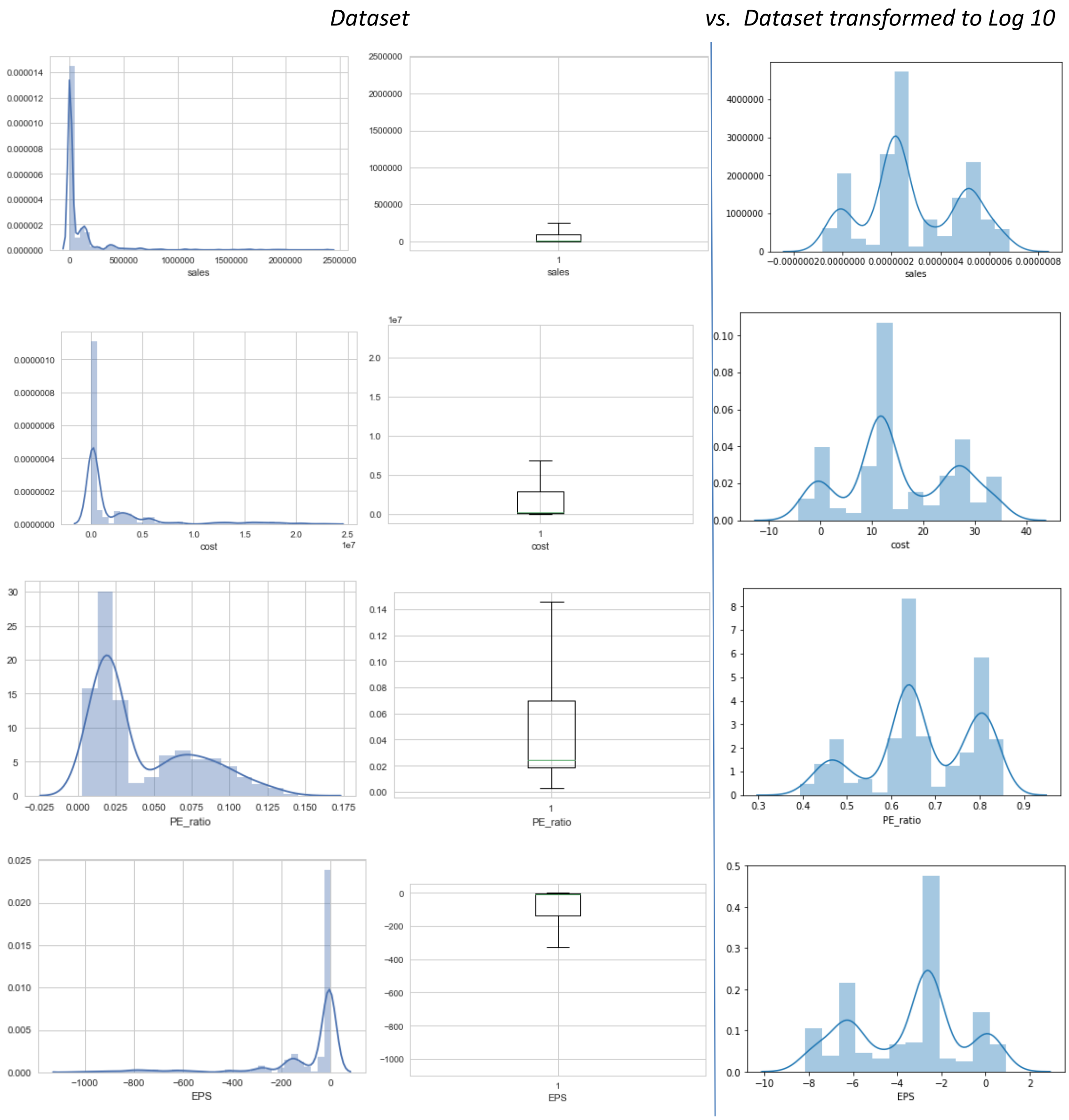


Figure 2.3- Distribution of the top valuations & ratios developed



Figures 2.2 and figures 2.3 illustrate how the top variable features of the network and the valuations/ratios metrics of the summary of the dataset are distributed. It indicates that the raw dataset is not normally distributed, has an irregular distribution (distribution skewed to the left and right) and extreme values (outliers). In addition, the price of ether is indicated to be approximately 50% of the time around 500. Every variable from the dataset has an extremely high standard deviation because the cryptocurrency market has high volatility. The behavior of all the variables from the dataset have similar behavior as the variables shown in figures 2.2 and 2.3, therefore, the variables are transformed to log 10 to achieve normal distribution and low standard deviation. The analysis of the distribution using histogram of the rest of the variables of the dataset is in appendix 1.

Figure 2.4 - Description (snapshot) of the variables of the dataset clean up transformed to log 10

	log10_eth_etherprice	eth_tx	eth_supply	log10_eth_marketcap	\
count	905.000000	905.000000	905.000000	905.000000	
mean	1.375784	4.830424	7.932633	3.308418	
std	0.944464	0.632444	0.038800	0.980971	
min	-0.376751	3.679155	7.862791	1.493901	
25%	0.921686	4.527140	7.897894	2.847616	
50%	1.087426	4.669884	7.936095	3.006540	
75%	2.407850	5.398929	7.969261	4.379289	
max	3.141456	6.130298	7.989807	5.127787	

	log10_eth_hashrate	log10_eth_difficulty	log10_eth_blocks	\
count	905.000000	905.000000	905.000000	
mean	3.925040	2.089254	3.733620	
std	0.864214	0.867100	0.071904	
min	2.522304	0.744136	3.451633	
25%	3.308842	1.435255	3.703205	
50%	3.793207	1.912031	3.770705	
75%	4.825619	3.029833	3.781037	
max	5.386574	3.481588	3.806790	

	log10_eth_uncles	log10_eth_blocksize	eth_gasprice	eth_gaslimit	\
count	905.000000	905.000000	905.000000	905.000000	
mean	2.663608	3.472721	10.467068	6.637215	
std	0.198638	0.504940	0.190846	0.176142	
min	2.100371	2.860338	10.015862	5.699177	
25%	2.547775	3.143639	10.353395	6.497207	
50%	2.622214	3.220892	10.380079	6.671993	
75%	2.700704	3.966517	10.618669	6.826641	
max	3.321391	4.527385	11.973128	6.903057	

	eth_gasused	log10_eth_ethersupply	CRIX_crypto_index	sales	\
count	905.000000	905.000000	905.000000	9.050000e+02	
mean	9.418847	4.430926	3.386932	2.863241e-07	
std	0.695295	0.085140	0.597909	1.970442e-07	
min	8.101485	4.171800	2.603296	-8.084372e-08	
25%	9.034792	4.389033	2.906700	1.914191e-07	
50%	9.215700	4.449943	3.125545	2.262162e-07	
75%	10.104995	4.500845	3.932011	4.974492e-07	
max	10.643100	4.562910	4.798618	6.799882e-07	

	cost	earnings	EPS	PE_ratio	price_per_sales	\
count	905.000000	905.000000	905.000000	905.000000	9.050000e+02	
mean	15.382640	-15.382639	-3.498363	0.670277	4.747574e+06	
std	10.476958	10.476958	2.433476	0.119231	3.972316e+05	
min	-4.179786	-35.251595	-8.154878	0.394052	0.000000e+00	
25%	10.446636	-25.771406	-6.012194	0.621293	4.708688e+06	
50%	12.334048	-12.334048	-2.736252	0.650149	4.816919e+06	
75%	25.771407	-10.446636	-2.324119	0.792252	4.829334e+06	
max	35.251596	4.179786	0.946564	0.853625	4.992082e+06	

	gasprice_per_trans	gasprice_per_blocks	hashrate_per_TPD	\
count	9.050000e+02	9.050000e+02	905.000000	
mean	5.499677e-08	7.687372e-08	0.803401	
std	3.397302e-08	5.344663e-08	0.081026	
min	-2.132445e-08	-2.186088e-08	0.626457	
25%	4.175486e-08	5.073550e-08	0.729451	
50%	4.896682e-08	5.995256e-08	0.819486	
75%	9.003404e-08	1.333065e-07	0.873842	
max	1.120054e-07	1.821268e-07	0.926638	

	blocksize_per_TPD	TPS
count	905.000000	905.000000
mean	0.718554	0.000057
std	0.030849	0.000007
min	0.678768	0.000043
25%	0.688962	0.000054
50%	0.716063	0.000055
75%	0.741577	0.000064
max	0.882223	0.000072

Figure 2.4 presents the cleanup of the log-dataset with the all the variables that were contemplated in the analysis.

Variables of the dataset

Figure 2.5 – Price and Crix Crypto Index

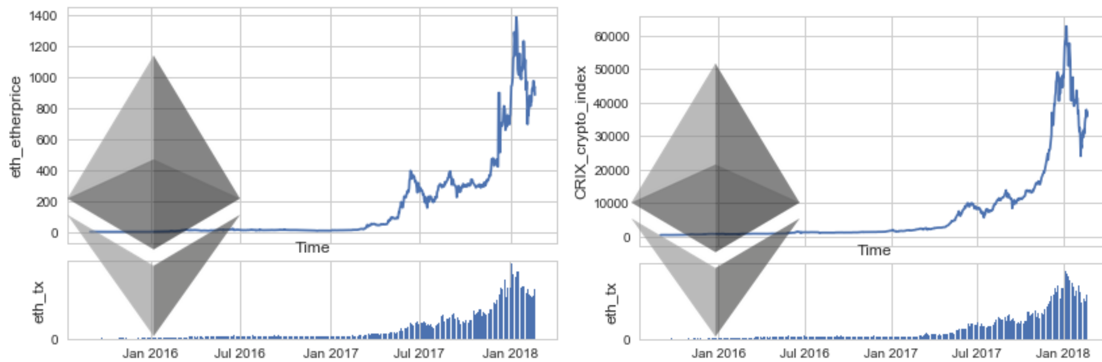


Figure 2.5 shows that the price of ether and Crix Crypto Index does not have a normal distribution, but is skewed to left and with extreme values. On the other hand, it can be seen that the Crix Crypto index behaves similarly to the price of ether, suggesting that the top 20 cryptocurrencies fluctuate like ether. It can be concluded then, that there is a strong relationship between ether and the market sentiment.

Key Features of the Ethereum Blockchain Network

Figure 2.6 - Key Features of the Network

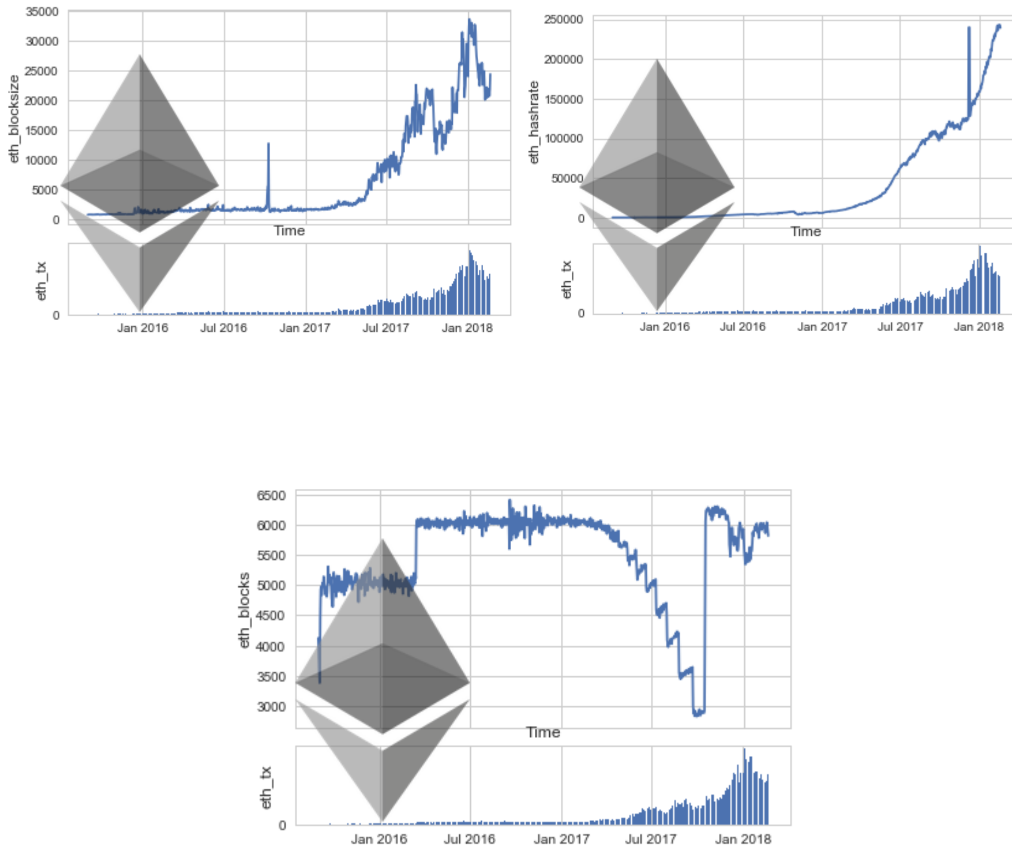


Figure 2.6 shows three key variables of the Ethereum Blockchain Network. It indicates that hash rate has been increasing substantially since 2017, which means that the speed of computing an operation (processing power) in Ethereum has increased. As a consequence, the health of the Ethereum Blockchain Network is improving through time. It can be seen that there is an outlier in hash rate, which could perhaps be the cause of the 51% of hack attack by miners.

On the other hand, blocksize increased and the number of blocks decreasing up to the end of 2017. It appears that Ethereum has been increasing the transactions per seconds (scalability) through increasing blocksize in order to allow more transactions to be input into a block. This is one approach to address the scalability issues of Blockchain. In fact, the main issues of the area of cryptocurrency that needs to be solved is this scalability issue.

Valuations metrics and ratios developed from the dataset of properties of the Ethereum Network (scalability, security and decentralization) and attributes of the Ethereum Blockchain (cost, speed, security).

Figure 2.7- Valuations/ratios of Cost

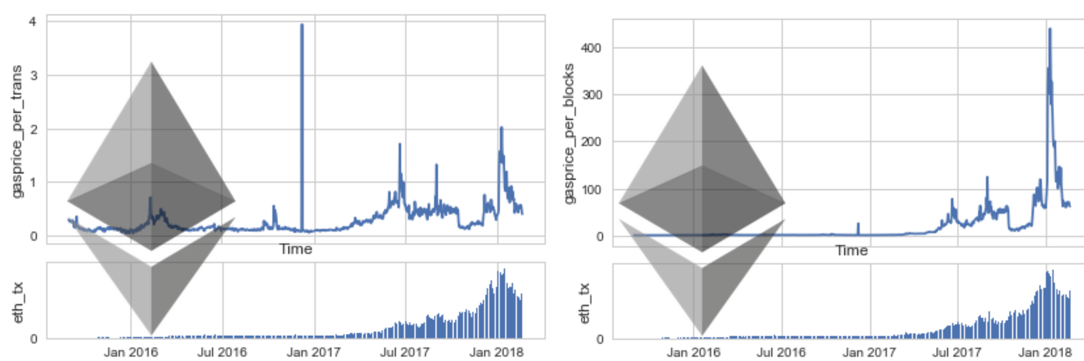


Figure 2.7 illustrates that the ratios of cost indicate that gas price per transactions and gas price fees per block of Ethereum are relatively constant. However, from 2017 there has been an increase in cost, which could be down to high demand. Again, there is an extreme value/outlier in gas price, believed to be caused by the 51% of hack attack by miners mentioned previously. It is worth noting that the gas price variable is the cost of transaction fees in micro ether.

Figure 2.8 – Valuations/ratios of security & decentralization

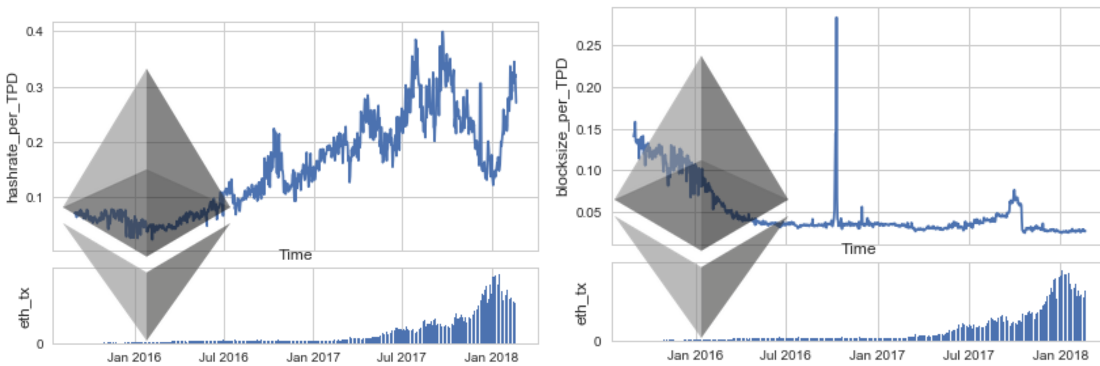


Figure 2.8 shows that hash rate per transaction has been increasing, with the exception of 2017 when there was a significant drop. It seems that this drop was due to a problem in the Ethereum Blockchain Network. On the other side blocksize per transaction is relatively constant with a slight decrease. Overall, it appears that the level of security and decentralization stay at normal levels.

Figure 2.9- Valuations of speed & scalability

Figure 2.9 illustrate that the scalability and speed of Ethereum network have been increasing up to an average of between 10 to 14 transactions per seconds until 2018. Again, these figure support the blocksize figures seen in 2.6, that through increasing the blocksize, the transaction per seconds can be increased. This places more transactions into a block. Indeed, some level of improvement in Ethereum in transaction per seconds is seen, with the average transactions per seconds increasing from 7 to 12 in 2017, suggesting that they are addressing the scalability issue.

Figure 2.10 Valuation/ratios of value

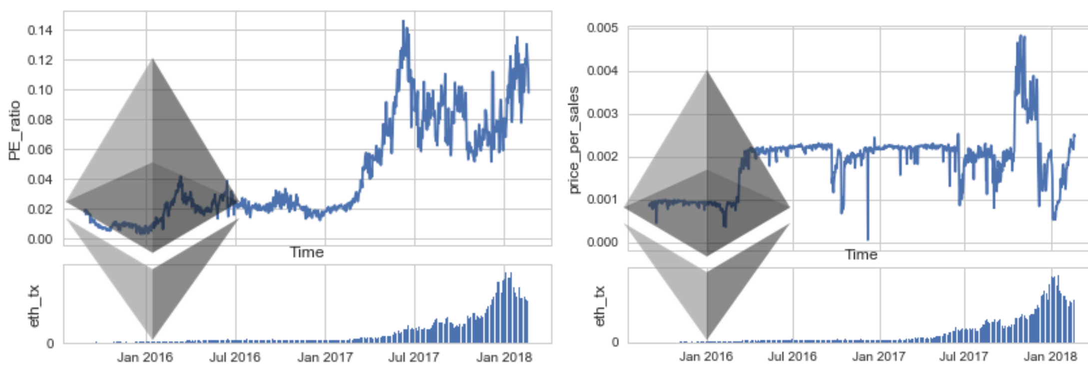
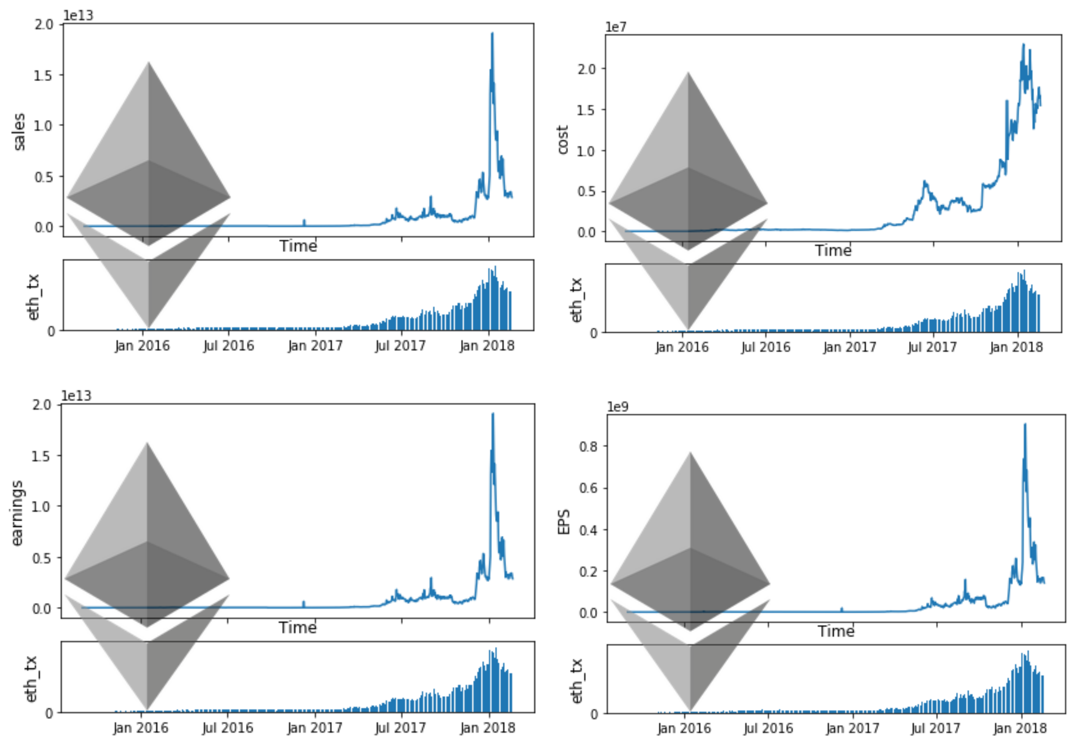


Figure 2.10 shows that investors from 2014 to 2017 increased their perspective about the valuation of ether because PE ratio increase. In fact, PE ratios is a proxy of value. In contrast, from 2017 to 2018 the PE ratio stayed relatively constant.

On the other hand, price to sales ratios reflect the same behavior as PE ratio, however a significant decrease occurred in 2017 in terms of valuation. It is inferred that the reason that these valuation ratios decreased last year was because of the decrease in the price of ether (as illustrated in figure 2.1). From 2017 the investors have been lowering their perspective about the value of Ethereum and have been selling Ethereum cryptocurrency.

In sum, both ratios reflect a decrease in value over the last year. One possible reason is the high levels of uncertainty about how to tackle the scalability issues with side chain, off chain or other approaches. In addition, uncertainly is created by the attempt to move from Proof of Work to Proof of Stake mechanism to tackle issues about electricity costs and other things.

Figure 2.11- Valuation/ratios of Value Sales, Cost, Earnings and EPS



- In order to better visualize the gap value the grid-lines are taken out from these figures
- Figures sales/transaction fees includes gas limit in the formula, in order to analyze the gap value between transaction fees and rewards for miners.

The valuations metrics named Sales (transaction fees) and Cost (reward to miners) are present to allow an analysis of the value of ether from the perspective of a business/company. Having said that, as seen in figure 2.11, after 2017 sales (transaction fees) have decreased and costs have increased. As a result, earnings and EPS of ether decreased. In other words, this indicates that the gap value of ether, which can be called Earnings or EPS, has decreased. It is observed that at the end of 2017 sales decreased more significantly than costs, and transactions decreased as well. It can be inferred that the main reason why earnings and EPS decreased in the end of 2017 is because people are using the Ethereum Blockchain network less (less demand) considering that transactions of ether have reduced.

Certainly, the intrinsic value of ether presents a substantial competitive advantage, which is the ability to build decentralized applications (Dapps). However, it is inferred that Ethereum has to address the issue of high mining costs in order to increase its value. This would make transaction fees cheaper and encourage more people to use the Ethereum blockchain Network.

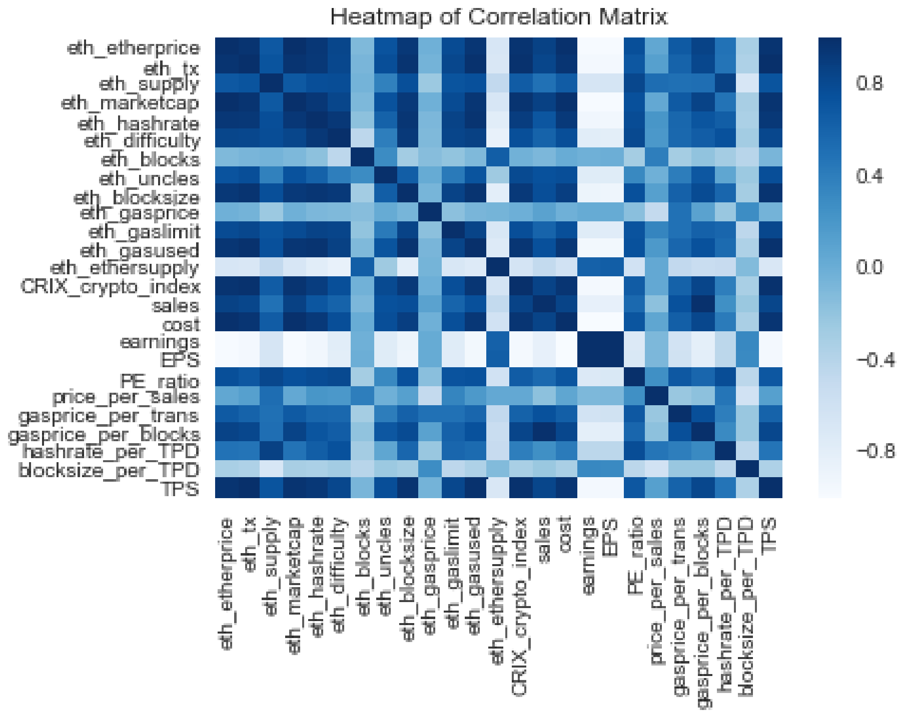
Correlation Matrix

Through the correlation matrix the relationship with price and other independent variables are observed. The correlation matrix is analyzed with the dataset and compared with the log-dataset. This identifies key variables that have more predicting power for price to perform the linear regression.

Figure 2.12- Correlation matrix with the dataset and log-dataset

Dataset *vs.* *Log dataset*

\	eth_etherprice		log10_eth_etherprice
eth_etherprice	1.000000	log10_eth_etherprice	1.000000
eth_tx	0.965446	eth_tx	0.983966
eth_supply	0.692380	eth_supply	0.938540
eth_marketcap	0.999913	log10_eth_marketcap	0.999907
eth_hashrate	0.941637	log10_eth_hashrate	0.974706
eth_difficulty	0.832706	log10_eth_difficulty	0.971325
eth_blocks	-0.098623	log10_eth_blocks	-0.134279 -
eth_uncles	0.732227	log10_eth_uncles	0.379100
eth_blocksize	0.930522	log10_eth_blocksize	0.944433
eth_gasprice	-0.022512	eth_gasprice	-0.467782 -
eth_gaslimit	0.794524	eth_gaslimit	0.711317
eth_gasused	0.952309	eth_gasused	0.970292
eth_ethersupply	-0.674280	log10_eth_ethersupply	-0.565421 -
CRIX_crypto_index	0.962146	CRIX_crypto_index	0.953653
sales	0.866710	sales	0.999600
cost	0.990381	cost	0.999036
earnings	-0.988506	earnings	-0.999036 -
EPS	-0.988179	EPS	-0.999238 -
PE_ratio	0.762463	PE_ratio	0.979971
price_per_sales	0.055379	price_per_sales	0.214718
gasprice_per_trans	0.677051	gasprice_per_trans	0.989351
gasprice_per_blocks	0.855860	gasprice_per_blocks	0.998414
hashrate_per_TPD	0.485784	hashrate_per_TPD	0.880138
blocksize_per_TPD	-0.313982	blocksize_per_TPD	0.094507
TPS	0.965446	TPS	0.983966


Heatmap of Correlation Matrix
eth_etherprice
eth_tx
eth_supply
eth_marketcap
eth_hashrate
eth_difficulty
eth_blocks
eth_uncles
eth_blocksize
eth_gasprice
eth_gaslimit
eth_gasused
eth_ethersupply
CRIX_crypto_index
sales
cost
earnings
EPS
PE_ratio
price_per_sales
gasprice_per_trans
gasprice_per_blocks
hashrate_per_TPD
blocksize_per_TPD
TPS
0.8
0.4
0.0
−0.4
−0.8

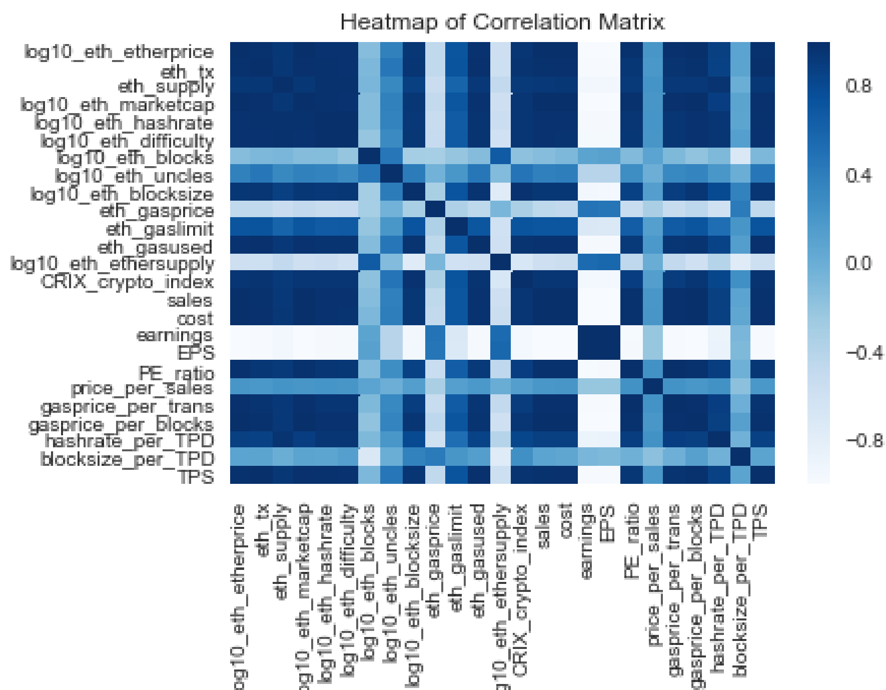

Heatmap of Correlation Matrix
log10_eth_etherprice
eth_tx
eth_supply
log10_eth_marketcap
log10_eth_hashrate
log10_eth_difficulty
log10_eth_blocks
log10_eth_uncles
log10_eth_blocksize
eth_gasprice
eth_gaslimit
eth_gasused
log10_eth_ethersupply
CRIX_crypto_index
sales
cost
earnings
EPS
PE_ratio
price_per_sales
gasprice_per_trans
gasprice_per_blocks
hashrate_per_TPD
blocksize_per_TPD
TPS
0.8
0.4
0.0
−0.4
−0.8

Figure 2.12 shows that the correlation matrix of the dataset and log-dataset have the same pattern of correlation. In fact, transforming the dataset to log 10 has a positive impact on the relationship of independent variables with price. For instance, it increases correlation of price with transactions, supply, gas price per transactions and hash rate per transaction. However, it has a negative impact because the features of the Ethereum Blockchain Network have more correlations against each other when delivering multicollinearity condition.

The strongest correlations between the price of ether and independent variables in the dataset are:

1- Cost (reward to miners)
2- Earnings
3- EPS
4- Transaction/seconds
5- Transaction
6- Crix Crypto Index
7- gas used
8- hash rate
9- blocksize
10- Sales

The strongest correlations between the price of ether and independent variables in the log-dataset are:

1. Sales (transactions fees)
2. EPS
3. Cost (reward to miners)
4. Earnings

5. Gas price per blocks
6. Gas price per transaction
7. Transaction/seconds
8. Transactions
9. PE ratio
10. Hash rate

In sum, the variables mentioned above will have more predicting power in the linear regression. The more correlated variables (without the log) are cost (reward to miners), earnings, and EPS. When these variables are transformed to log 10, the high correlation remains, adding the variable of Sales (transaction fees) to the most significant variables in the correlation matrix. This indicates that transaction fees and rewards for miners are the most significant variables for price.

In addition, in the dataset without log 10, the strongest positive correlation (moving in the same direction) are identified with the variables of cost, transactions per seconds, transactions, Crix Crypto Index and hash rate. The variables that have the strongest negative correlation are Earnings, EPS, Supply and blocksize. This dataset is intuitive and logical because when transactions and the Crix Crypto Index increase, the price of ether increases, and if supply decrease, the price of ether increases. Certainly, these variables fluctuate like this in the real market of cryptocurrencies.

The only variable that has relatively no correlation is price per sales variable in the dataset. However, when the variable of the price of sales is transformed to log 10, its correlation with price increases. In summary, the importance of taking every variable into account is evident, from the model of linear regression to the log-dataset, because they all have a correlation with price.

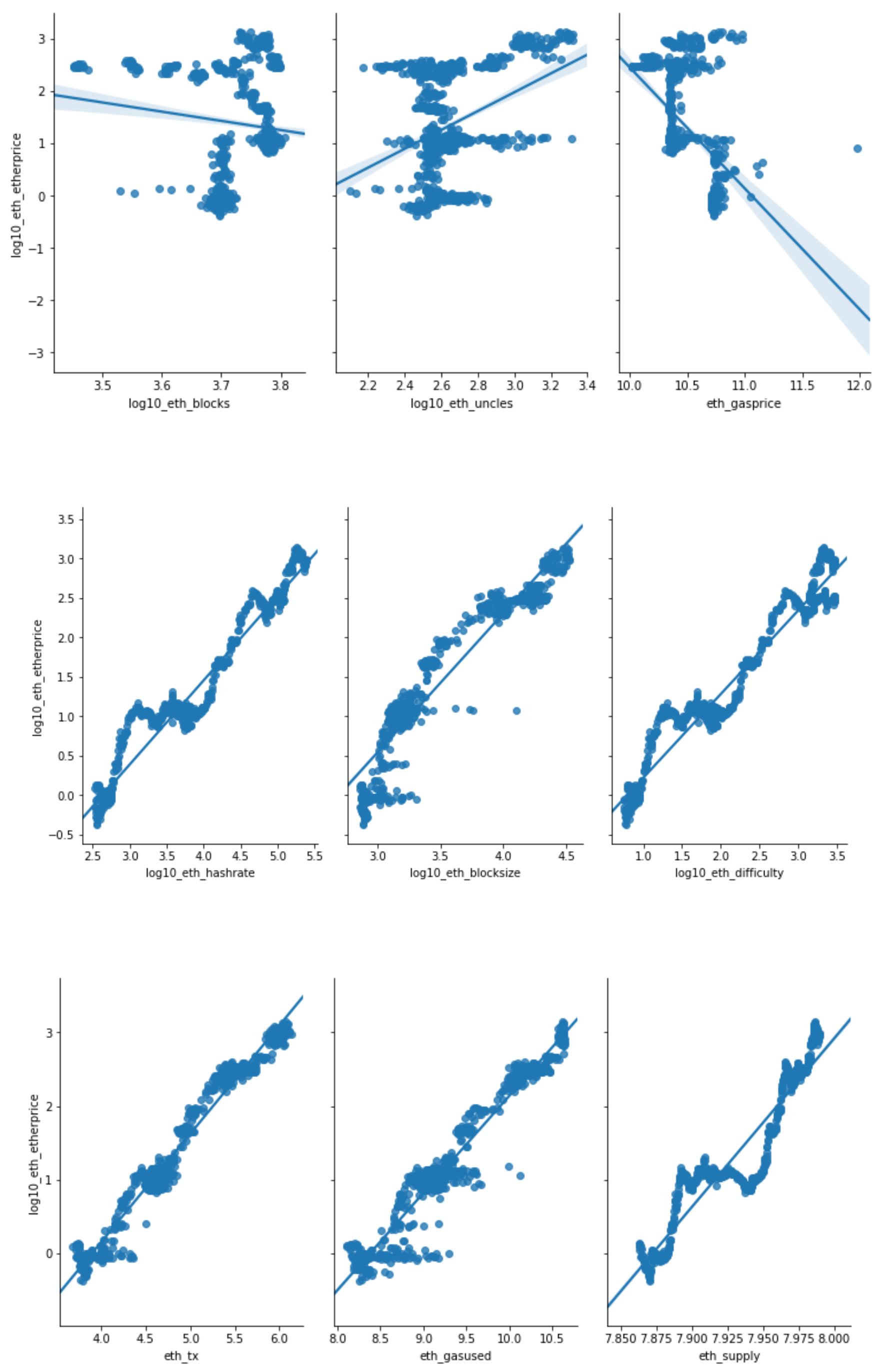
Assumptions about Linear Regression

Supervised methods, such as linear multiple regression, were utilized to be able to predict the dependent variable (price) with the independent variables. Linear regressions have the following assumptions:

- There is a linear relationship between price and independent variables – All the variables meet the assumption to a certain level.
- Errors between the model and the observed values are normally distributed
- Variance of the error is constant with a mean of 0.
- Errors are independent from each other.
- Multicollinearity condition: there are high degrees of multicollinearity among the independent variables
- Outliers: variables with extreme values are transformed to log 10 to achieve symmetrical distribution.
- Sample size: the dataset is large enough to meet the assumption. It has 902 daily prices of ether (rows) and 26 variables (columns), including time.

To see if the assumptions of linear relationship and multicollinearity condition are met, they are analyzed through scatterplots.

Figure 2.13 - Price against features of the Network to assess assumption of linear relationship



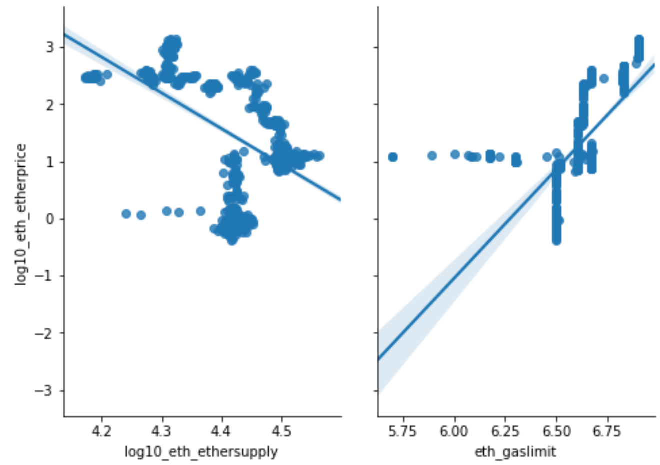
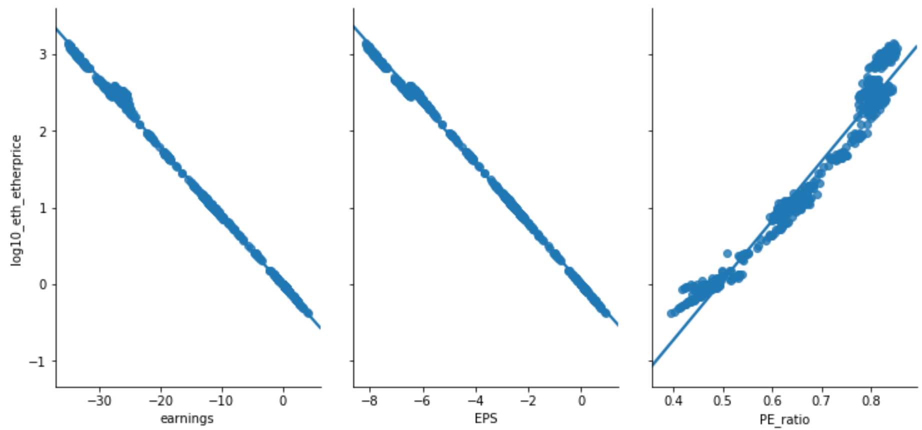


Figure 2.13 illustrates clearly that independent variables of the features of the Network from the dataset have some level of linear relationship with price. It highlights that the variables hash rate, transactions and gas used, have the most clear lines suggesting that these variables will deliver more predicting power to the linear regression.

Figure 2.14 - Scatterplot of price against top valuations/ratios, developed to assess assumption about the linear relationship



On the other hand, three valuations metrics are selected, in order to see more clearly how much predicting power is going to be delivered to the linear regression. Figure 2.14 shows that have the strongest linear correlation with price. It is inferred then, that the variables of valuations/ratio compared to the features of the Network will deliver more predicting power. Therefore, building this valuations metrics and taking into account the PE ratio developed by Woo (2017), will help deliver extra value and will support a superior model.

Figure 2.15 – Price against blocksize per transaction, price of sales and uncles to assess the assumption of linear relationship

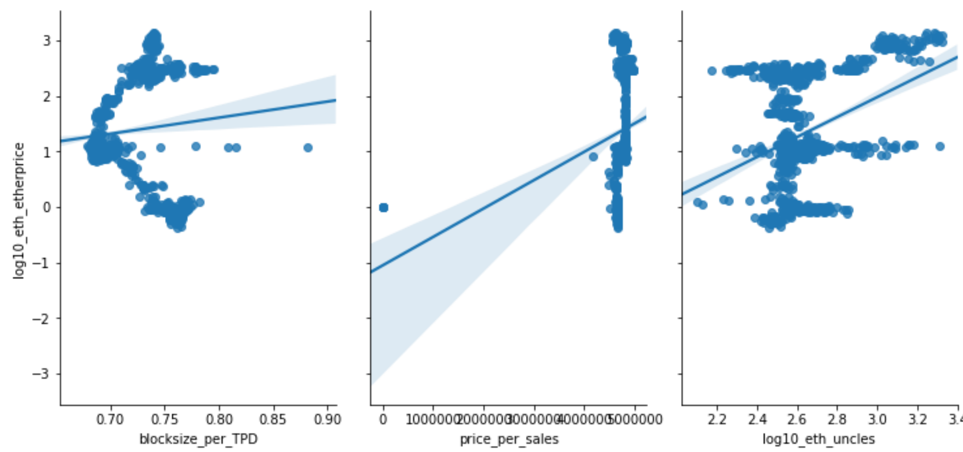
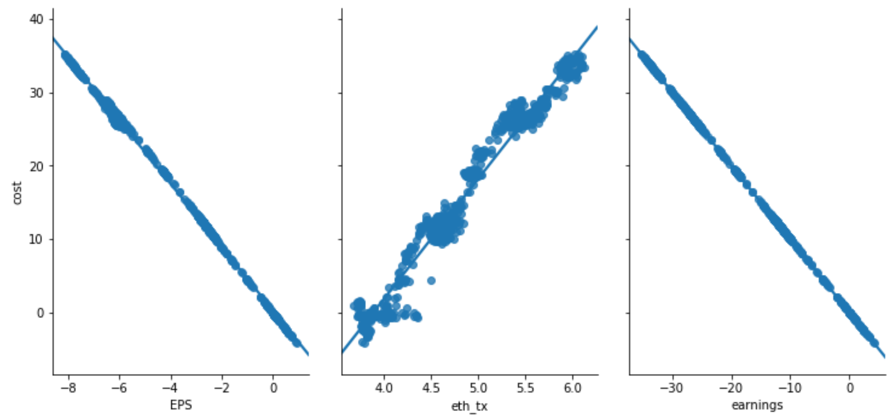


Figure 2.15 selects the variables in the log-dataset with the lowest correlation with price, in order to visualize more clearly/closer the linear relationship. These variables are blocksize per transaction, price per sales and uncles. Figure 2.15 shows that even though these variable have a relatively low correlation to price, there is still a relatively small degree of linear relationship. Thus, this information contributes to predicting the price of ether to a small extent, which is why the linear regression is taken into account.

Figure 2.16 – Cost and Sales variables assessing the condition of multicollinearity



The cost variable (reward to miners) is selected because it is the variable with the highest correlation with price. This shows the multicollinearity condition of the highly correlated independent variables. Figure 2.16 illustrate that the variable of cost delivers a strong multicollinearity condition because it has a strong linear relationship with other independent variables such as EPS, transactions and earnings. Thus, the assumption of no condition of multicollinearity in this log dataset is not met. In fact, not meeting the assumption of no the multicollinearity condition can affect the direction of the coefficient in the linear regression. Hence, when the linear regression is performed it has to verify if the coefficients (positive or negative) are logical.

Linear Regression

Initially, the linear regression is performed with 23 independent variables that have linear relationship with the dependent variable (price). Secondly, cross validation is done, and the dataset is split between 80% for training set and 20% for test set. After all the assumptions are almost met and the data is split, the linear regression is performed.

In order to evaluate the accuracy of the model and compare which dataset fitted the model better (dataset or the log-dataset), evaluation metrics such as Mean Square Error, R square and variance of the predicted values are performed.

Figure 2.17 – Mean Squared Error, R Squared and Variance of the dataset and log-dataset of the linear regression

Log-dataset

```
In [208]: print("MEAN SQUARED ERROR: ", metrics.mean_squared_error(Y_test, predict))
MEAN SQUARED ERROR:  6.680611290104756e-07

In [209]: print("R2 SCORE: ", metrics.r2_score(Y_test, predict))
R2 SCORE:  0.9999993157991842

In [210]: predict.var()
Out[210]: 0.9763327132128988
```

Dataset

```
In [265]: print("MEAN SQUARED ERROR: ", metrics.mean_squared_error(Y_test, predict))
MEAN SQUARED ERROR:  35.290813395003866

In [266]: print("R2 SCORE: ", metrics.r2_score(Y_test, predict))
R2 SCORE:  0.9995671057196319

In [267]: predict.var()
Out[267]: 81821.72538058205
```

Figure 2.17 indicates that Mean Squared Error is extremely low with 6.6806*e^-7. This explains why the model has a small degree of error regarding the differences between predicting values and actual values. In addition, R Square is 99%, indicating that the model explains the actual values strongly. The predicted values that have a variance of .97, meaning that the predicted values have high volatility.

In contrast, the dataset without the log has Mean Squared Error of 35.2908. Indicating that there is a higher degree of error in the model, in comparison with the log-dataset model, but it has a high R Square of 99%. Moreover, the variance of the predicted values of the dataset is extremely high. Therefore, the log-dataset fitted the model better because it has a significantly smaller degree of error (MSE) and, secondly, the variance of the prediction values is extremely low compared to the log-dataset.

The model with the log-dataset has outstanding results, believed to be because it considered all relevant factors/variables that can affect the price of ether. Hence, the dataset that is considered for the model is the log-dataset.

Figure 2.18 – Intercept and coefficients of the linear regression with the log-dataset

```
In [206]: print("lm.intercept_", lm.intercept_)
lm.intercept_ 0.6186321280499227
```

```
In [212]: print (L)
    Independent variables  Coefficients
0                  eth_tx -2.255166e-02
1              eth_supply -7.961761e-02
2       log10_eth_hashrate  4.227367e-02
3     log10_eth_difficulty -1.141742e-02
4         log10_eth_blocks -5.541146e-02
5         log10_eth_uncles -3.644946e-03
6      log10_eth_blocksize  3.536289e-02
7            eth_gasprice  1.512719e-03
8            eth_gaslimit -2.082513e-04
9             eth_gasused  4.335297e-06
10  log10_eth_ethersupply  4.272529e-02
11       CRIX_crypto_index  3.030525e-03
12                  sales -2.684979e+06
13                   cost -1.342497e+06
14               earnings -1.342497e+06
15                    EPS  1.066549e-04
16               PE_ratio  2.355343e-01
17         price_per_sales  1.646558e-11
18      gasprice_per_trans -7.390942e+05
19     gasprice_per_blocks  1.546419e+07
20        hashrate_per_TPD -1.412013e-01
21       blocksize_per_TPD -1.657474e-01
22                    TPS -1.760925e-05
```

The figure 2.18 demonstrates the top 10 variables that have more predicting power in the linear regression in order of significance, from the largest to smallest:

1. Gas price per blocks
2. Sales (transactions fees)
3. Cost (reward to miners)
4. Earnings
5. Gas price per transaction
6. PE ratio
7. Blocksize per transaction
8. Hash rate per transaction
9. Supply
10. Blocks

The results from the linear regression are related to the correlation's matrix results because they again highlight that data related to transactions fees and rewards to miners has the most predicting power for the price of ether.

It is worth noting that this model indicates that the variable gas price, which represents the transaction fees in micro ether; when gas price is combined with other variables (developing ratios), it becomes the most significant variable to the price of ether. This means that, at the end of the day, the most important variable for predicting the price of ether is the transaction fees.

In fact, the coefficient in the linear regression shows how the price of ether changes when the value of the independent variable changes. As an example of this, the coefficient of gas price per transaction is -7.39 *e^5 meaning that by a change in one unit of gas price per transaction there will be -7.39 *e^5 of change in the value of the price of ether. This is intuitive because the transaction fee per transaction decreases and as a result the price of ether increases, due to the Ethereum Blockchain Network being cheaper.
On the other hand, the cost variable, which is the reward to miners, increases and as a result the price of ether will decrease -1.34 e^6. This can also be seen as intuitive because reducing the cost of building a block in Ethereum makes the network more valuable because it becomes cheaper to produce a block. There is an indication then, that if Ethereum focused specifically on the attributes of blockchain, in particular the cost, reducing transaction fees and rewarding miners will lead to substantial increases in the value of ether.

Certainly, nowadays one of the main issues with the blockchain technology, that is being attempted to be solved as a priority, is the mining cost of the Proof of Work mechanism. Currently, Ethereum wants to transfer to a Proof of Stake mechanism, which would lead to transactions fees being reduced. As soon this issue is solved, the value of ether will increase substantially, according to the linear regression. Hence, from an investor's perspective, now would be a good time to invest because when the Proof of Stake Mechanism is incorporated (possibly next year), the intrinsic value of ether could increase substantially.

The most significant variables to the price of ether in the linear regression are the cost of the Network followed by valuation metrics like Earnings and PE ratio. The next most significant variables are the key features of the Network such as blocksize and hash rate. It can be inferred by the coefficient of the Crix Crypto Index, which represents the market sentiment, which has less predicting power to the price of ether than the features of the Network. It seems that these results are strongly meaningful for the cryptocurrency area because they indicate that market sentiment influences the price of ether, however, it is not concluded to be one of the top influences, as some people believe.

The results suggest that the gas price (transaction fees) for building a block on the Ethereum Blockchain Network with a specific blocksize and hash rate, are fundamental variables of the Network that have more influence on the price. Therefore, it can be concluded that features of the Network and the functionality features of Ethereum have more predicting power than market sentiment. Thus, it is believed that the fundamental analysis of Ethereum is more important than technical analysis if the aim is to arrive at an understanding of the intrinsic value of ether.

Proposed Model

Though the Linear Regression has outstanding results, it can over-fit the data, because of the high amount of independent variables. Having said that, in order to address the issue of over-fitting it is possible to re-fit the linear regression to find the strongest coefficient for the model. Therefore, the linear regression is performed (with cross-validation) involving the top 9 coefficient variables of linear regression adding one by one, in order to achieve the best trade-off between the number of variables and Mean Square Error.

Table 2.2 – Trade-off between top 9 strongest coefficient variables in the linear regression and Mean Square Error

Trade-off between # variables and MSE

# Variable		MSE	R Squared	Variance of predicted values
1		2.6049E-03	0.9969	85.28%
2		5.4084E-04	0.9994	86.68%
3	Cut-off	1.1266E-06	0.9999	92.84%
4		1.0010E-06	0.9999	81.51%
5		8.0977E-07	0.9999	82.39%
6		7.4011E-07	0.9999	87.50%
7		9.9574E-07	0.9999	88.74%
8		8.7621E-07	0.9999	80.06%
9		9.1779E-07	0.9999	90.52%

Table 2.2 shows an increase in the number of independent variable in the linear regression, and a decrease in the Mean Squared Error (MSE). The best trade-off of number of variables and MSE is identified as being with the top three variables with the strongest coefficient of the linear regression, which are gas price per block, sales (transactions fees) and cost (reward to miners). Therefore, the cut-off is in the top three variables of strongest coefficient of the linear regression because between selecting 2 and 3 variables is a significant difference in the MSE; after selecting 3 variables or more

variables of the linear regression the MSE is relative constant. Meaning there is not a significant difference with errors between predicted values and actual values. Thus, adding more than three variables in the linear regression does not add significant predicting power to the model. Hence, a superior model for predicting the price is proposed with the aim of achieving exceptional returns with these three variables.

Figure 2.19 - Results proposed model of ether (MSE, R Square, Variance, Intercept, Coefficients)

```
In [146]: print("MEAN SQUARED ERROR: ",
metrics.mean_squared_error(Y_test, predict))
MEAN SQUARED ERROR:  1.126644977022898e-06

In [147]: print("R2 SCORE: ", metrics.r2_score(Y_test,
predict))
R2 SCORE:  0.9999987868170999

In [148]: predict.var()
Out[148]: 0.9284679038018444
```

```
In [150]: print("lm.intercept_", lm.intercept_)
lm.intercept_ -0.0001525705520144527
```

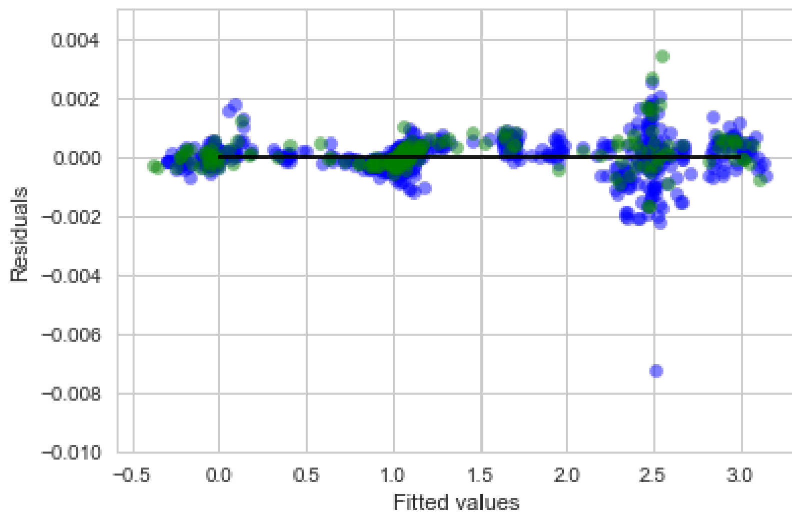
```
  Independent variables   Coefficients
0   gasprice_per_blocks   1.670198e+07
1                 sales  -4.384733e+06
2                  cost   8.759393e-02
```

The Proposed model is the following:

*Predicted price of ether = -0.0001525 + (1.6702e^07) * gas price per blocks + (-4.3847e^06) * sales/transaction fees + (8.7594e^-02) * cost/reward to miners*

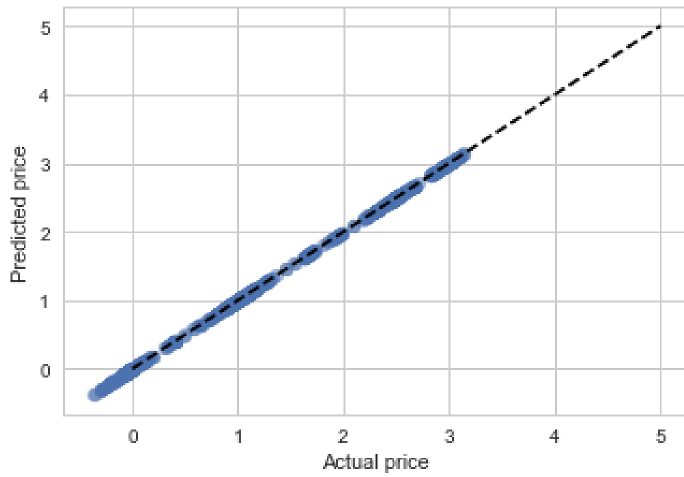
The proposed model is intuitive because if transaction fees are increased, the price of ether will decrease and if the reward for miners is increased, the price of ether will increase. In addition, if gas price per blocks increases, the price of ether increase. This is logical because transaction fees are there to pay miners.

Figure 2.20 – Residuals of the proposed model



In order to analyze if the proposed model met the assumptions about residuals (a mean of 0 and constant standard deviation) an illustration is performed. In figure 2.20 the residuals are centered to 0, which mean the residuals are closest to the line, and has constant variance because residuals are plotted constantly on both sides of the line. It is inferred then, that the proposed model supports the assumptions of the residuals.

Figure 2.21 - Compare predicted price of the proposed model and actual price



Finally, Figure 2.21 visualizes the prediction of the proposed model with test log-dataset and shows that the predicted prices are really close the actual prices. This indicates that the proposed model has strong power to predict the price of ether highly accurately. However, considering the high variance in the predicted prices, it is correct to conclude that the ether cryptocurrency is most suited to investors with an appetite for higher risks.

Findings/Implications

The key insight that is derived from this research is that investor in the cryptocurrency space have a misconception that market sentiment is the strongest variable for predicting the price of ether. The results from the model support to some extent, the mindset of investors that market sentiment is the key factor that drives the price of cryptocurrency, but this is not the case for the Ethereum Blockchain Network. It seems that this is due to the functionality feature and the utility of ether. Furthermore, the valuations metrics, such as the cost, value and features of the Network have more predicting power than market sentiment. Therefore, the Ethereum Blockchain Network is seen to behave like a traditional company, meaning that the intrinsic value of ether is supported by fundamental analysis rather than technical analysis. It appears then, that cryptocurrency is no longer an area for speculators.

The implication of this model is that Ethereum works with the Proof of Work mechanism. Perhaps then, it is possible to infer that this model can perform with other cryptocurrencies that work with different consensus mechanisms like cryptocurrency EOS, which use Proof of Stake mechanisms. Another implication is that this model can perform for other cryptocurrencies that have Proof of Work mechanisms that do not have the functionality feature or the same utility as ether. Bitcoin, for instance does not have the same functionality feature or utility, but could perhaps perform well with this model.

These results contribute to the body of knowledge and academic research, by identifying and developing ratios/functions in the linear regression, increase the predicting power of variables in the area of cryptocurrency. Key to this were the variables developed from the dataset like gas price per blocks, gas price per transaction and Earnings. The study has also been able to identify the variables with the most power to predict the price of ether. The study can, therefore, be seen to contribute to the body of knowledge in that these variables have been identified as significant of deeper analysis in the future.

In practice this research means that applying the model will add extra value for forecasting the price of cryptocurrencies through the Proof of Work mechanism. This is because it is possible to compare other methods' prediction prices, taking the mean from the multiple methods and delivering a more accurate prediction of ether. Thus, the estimation of the price of the cryptocurrency will be supported by multiple models, which is one of the practical processes of valuation. Secondly, the results of the proposed model are noteworthy, so that applying this model to other cryptocurrency consensus mechanisms and other cryptocurrencies that use a Proof of Work mechanism and have a different utility, could produce similar results.

The proposed model contributes to the fintech companies in the area of asset management (robot-advisor) because of its simplicity and notable results regarding predicting power, that could achieve outstanding returns for clients. This model could also be used as a benchmark to compare with traditional models. In addition, this model can help bring the attention to the new fintech companies in the asset management side, where it can be delivering the significant variables of the proposed model to their own models. Chiefly, fintech companies can change their approach to analyzing prices of cryptocurrencies to a more fundamental analysis that concentrates on valuations metrics and ratios.

In sum, this research can change the mindset theoretically, practically and of how fintech companies analyze cryptocurrencies, by changing the focus of attention away from technical analysis towards fundamental analysis.

Conclusion

In conclusion the main contribution of this research to the area of cryptocurrency is the proposed model for predicting ether prices for investors, which demonstrates that fundamental analysis of the intrinsic value of cryptocurrency can help deliver extraordinary investment returns compared to technical analysis. The model highlights that the most significant variables to the value of ether are gas price to blocks, transactions fees and rewards to miners. Certainly, with the valuations metrics and ratios delivered in this research, cryptocurrencies can be analyzed more deeply. Thus, it is believed that after the results of the proposed model, the functionality feature of ether and utility indicate that it is best to analyze cryptocurrency from the perspective of a business.

Reflecting on the complications of the research. The key complications identified were related to the dataset, where missing values were encountered. In addition, there were further complications related to the metrics of the features of the Ethereum Network, like the number of nodes, not being available. Hence, the quality of the dataset is not completely robust. To solve this issue of missing values, the average value of the last 10 observations (this happen only three times) are taken into account. Furthermore, the first two months of the dataset contained a large amount of missing values. For this reason these months of data were excluded from the analysis.

It is important to recognize that this research this is just a starting point and can be viewed as the beginning of deep fundamental analysis into the area of cryptocurrency.

Limitations of the Research

The main limitation of this research is that the proposed model is applied to only one cryptocurrency (ether), which works with a Proof of Work mechanism.

Future Projections

There is potential for future research focusing on how the valuations metrics and ratios behave in other cryptocurrencies, using this research as a comparison. It would also be possible to apply the proposed model to analyzing other cryptocurrencies with Proof of Work mechanisms and different utility. Another area for future research could be to analyze cryptocurrencies that have different consensus mechanisms, like Proof of Stake, in order to find out if the model performs well and can be applied with these. Finally, evaluating how these other cryptocurrencies' valuations metrics and ratios contribute to the predicting power of the proposed model, could be an area for further investigation.

Bibliography

Amjad, M. and Shah, D. (2017). *Trading Bitcoin and Online Time Series Prediction*. Massachusetts Institute of Technology.

Bloomberg.com. (2018). *Bloomberg and Galaxy Digital Capital Management Launch Cryptocurrency Benchmark Index*. [online] Available at: https://www.bloomberg.com/company/announcements/bloomberg_galaxy_launch_cryptocurrency_index/ [Accessed 30 Jun. 2018].

Brown, A. (2017). *How to Make Sense of Cryptocurrency Valuations*. [online] Bloomberg.com. Available at: https://www.bloomberg.com/view/articles/2018-04-17/how-to-make-sense-of-cryptocurrency-valuations [Accessed 1 Jul. 2018].

Chen, M., Narwal, N. and Schultz, M. (2017). *Predicting Price Changes in Ethereum*. Stanford University.

De, N. (2018). *SEC Official Pushes Back on Claims Ether Is a Security - CoinDesk*. [online] CoinDesk. Available at: https://www.coindesk.com/sec-official-ether-is-not-a-security/ [Accessed 16 Jul. 2018].

Greaves, A. and Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. Stanford University.

Guo, T. and Antulov-Fantulin, N. (2018). *An experimental study of Bitcoin fluctuation using machine learning methods*. ETH Zurich (Swiss Federal Institute of Technology in Zurich).

Kaggle.com. (2018). *Cryptocurrency Historical Prices | Kaggle*. [online] Available at: https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory/data [Accessed 3 Jun. 2018].

Katz, L. (2018). *Bitcoin Bull Tom Lee Goes Hyperbolic on Latest Price Forecast*. [online] Bloomberg.com. Available at: https://www.bloomberg.com/news/articles/2018-01-18/bitcoin-bull-tom-lee-goes-hyperbolic-on-latest-price-forecast [Accessed 1 Jul. 2018].

Lamon, C., Nielsen, E. and Redondo, E. (2017). *Cryptocurrency Price Prediction Using News and Social Media Sentiment*. Stanford University.

Madan, I., Saluja, S. and Zhao, A. (2014). *Automated Bitcoin Trading via Machine Learning Algorithms*. Stanford University.

Raza, S. (2014). *A Closer Look At Warren Buffett's 'Price Is What You Pay, Value Is What You Get' Quote*. [online] ValueWalk. Available at: https://www.valuewalk.com/2014/09/warren-buffett-famous-quotes/ [Accessed 4 Jul. 2018].

Shah, D. and Zhang, K. (2014). *Bayesian regression and Bitcoin*. Massachusetts Institute of Technology.

Sorgente, M. and Cibils, C. (2014). *The Reaction of a Network: Exploring the Relationship between the Bitcoin Network Structure and the Bitcoin Price*. Stanford University.

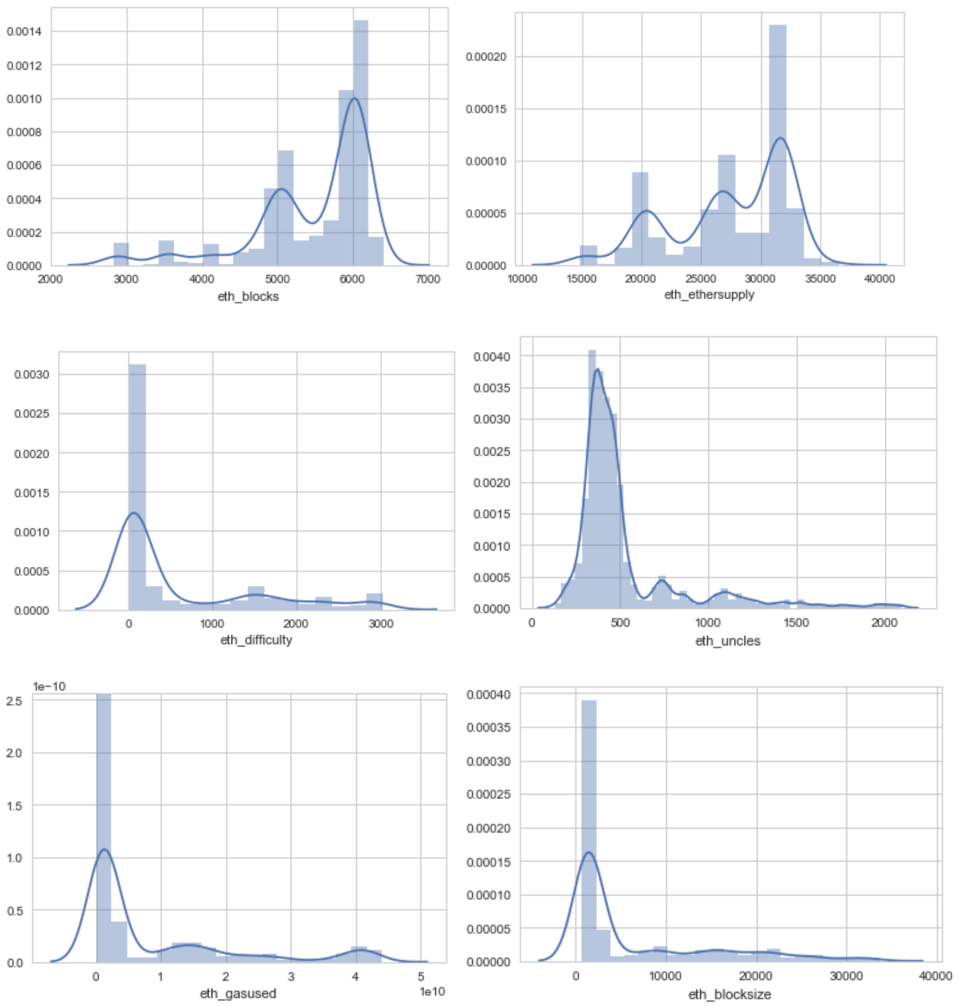
Trimborn, S. and Härdle, W. (2016). *CRIX or evaluating blockchain based currencies*. Humboldt University of Berlin.

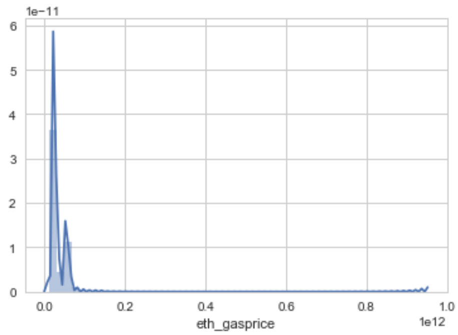
Tsaklanos, T. (2017). *An Ethereum Price Forecast Of $1000 - Investing Haven*. [online] Investing Haven. Available at: https://investinghaven.com/screening/ethereum-price-forecast-1000/ [Accessed 1 Jul. 2018].

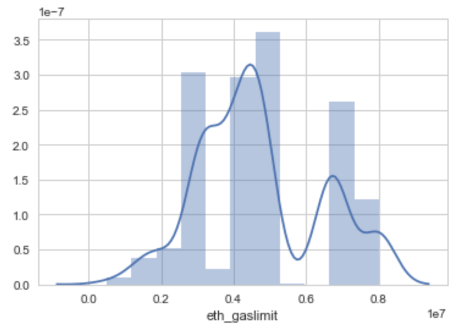
Woo, W. (2017). *Bitcoin NVT Ratio : Woobull Charts*. [online] Charts.woobull.com. Available at: http://charts.woobull.com/bitcoin-nvt-ratio/ [Accessed 3 Jul. 2018].

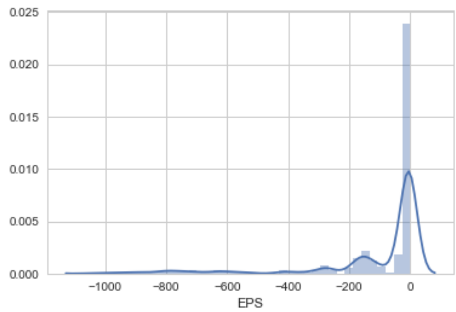
Appendix

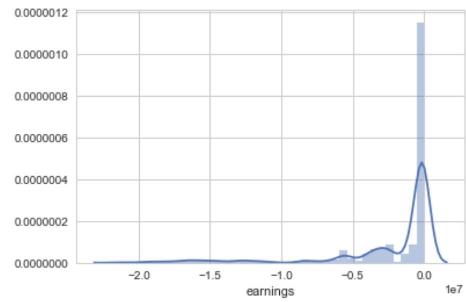
Appendix 1 - Distribution of the variables not illustrate in the research

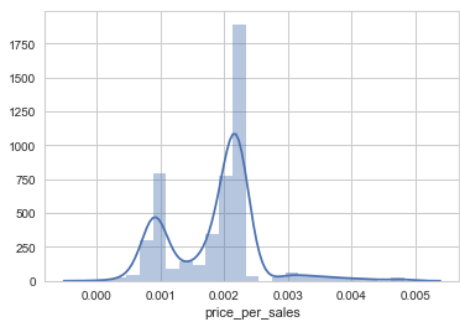


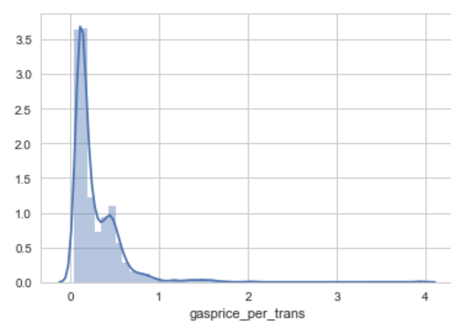

1e−11
eth_gasprice
1e12

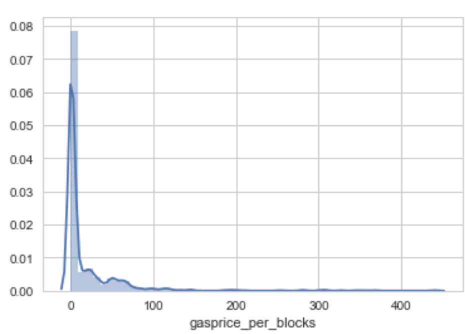

1e−7
eth_gaslimit
1e7

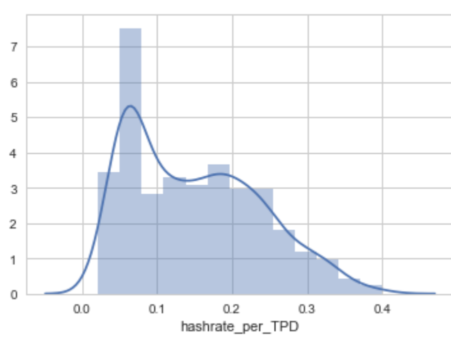

EPS

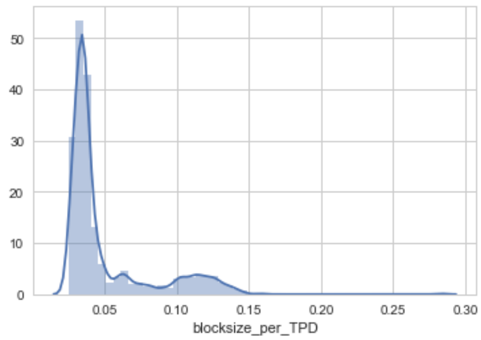

earnings
1e7

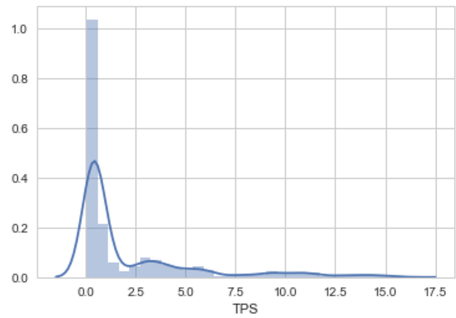

price_per_sales


gasprice_per_trans


gasprice_per_blocks


hashrate_per_TPD





Appendix 2 – Code

```
import datetime

import sys

import io

import sklearn

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.dates as mdates

import seaborn as sns

from PIL import Image

from sklearn.linear_model import LinearRegression

from sklearn import metrics



# Console settings to show full results
```

```
pd.set_option('display.max_columns', 500)

pd.set_option('display.expand_frame_repr', True)



# Create and clean original dataset

# Importing dataset

#Crix Crypto Index

crixDataset = pd.read_json("/Users/joshuaeick/Desktop/Joshua Eick/crix.json")

#Ethereum

dataset = pd.read_csv("/Users/joshuaeick/Desktop/Joshua
Eick/ethereum_dataset_kaggle.csv")



#Cleaning the dataset

#drop 4 columns/variables not important for the research

dataset = dataset.drop(columns=['eth_ens_register', 'UnixTimeStamp', 'eth_address',
'eth_blocktime'])

#drop the first 32 rows because have to much missin values

dataset = dataset[32:]

#reseting the index

dataset = dataset.reset_index()

dataset = dataset.drop(columns='index')
```

```
#Adding more columns to the dataset of valuations metrics/ratios and index

dataset['CRIX_crypto_index'] = 0.0

dataset['Date(UTC)'] = pd.to_datetime(dataset['Date(UTC)'])

dataset['sales'] = 0

dataset['cost'] = 0

dataset['earnings'] = 0

dataset['EPS'] = 0.0

dataset['PE_ratio'] = 0.0

dataset['price_per_sales'] = 0.0

dataset['gasprice_per_trans'] = 0.0

dataset['gasprice_per_blocks'] = 0.0

dataset['hashrate_per_TPD'] = 0.0

dataset['blocksize_per_TPD'] = 0.0

dataset['TPS'] = 0.0


# Change 'eth_tx' to a floating point number to get more accurate results

dataset['eth_tx'] = dataset.eth_tx.astype(float)


# Create log-dataset

logDataset = pd.DataFrame()
```

```
logDataset['Date(UTC)'] = (dataset['Date(UTC)'])

logDataset['log10_eth_etherprice'] = np.log10(dataset['eth_etherprice'])

logDataset['log10_eth_etherprice'] = logDataset.log10_eth_etherprice.astype(float)

logDataset['eth_tx'] = np.log10(dataset['eth_tx'])

logDataset['eth_supply'] = np.log10(dataset['eth_supply'])

logDataset['log10_eth_marketcap'] = np.log10(dataset['eth_marketcap'])

logDataset['log10_eth_hashrate'] = np.log10(dataset['eth_hashrate'])

logDataset['log10_eth_difficulty'] = np.log10(dataset['eth_difficulty'])

logDataset['log10_eth_blocks'] = np.log10(dataset['eth_blocks'])

logDataset['log10_eth_blocks'] = logDataset.log10_eth_blocks.astype(float)

logDataset['log10_eth_uncles'] = np.log10(dataset['eth_uncles'])

logDataset['log10_eth_blocksize'] = np.log10(dataset['eth_blocksize'])

logDataset['eth_gasprice'] = np.log10(dataset['eth_gasprice'])

logDataset['eth_gaslimit'] = np.log10(dataset['eth_gaslimit'])

logDataset['eth_gasused'] = np.log10(dataset['eth_gasused'])

logDataset['log10_eth_ethersupply'] = np.log10(dataset['eth_ethersupply'])

logDataset['CRIX_crypto_index'] = 0.0

logDataset['sales'] = 0.0

logDataset['cost'] = 0.0

logDataset['earnings'] = 0.0
```

```
logDataset['EPS'] = 0.0

logDataset['PE_ratio'] = 0.0

logDataset['price_per_sales'] = 0.0

logDataset['gasprice_per_trans'] = 0.0

logDataset['gasprice_per_blocks'] = 0.0

logDataset['hashrate_per_TPD'] = 0.0

logDataset['blocksize_per_TPD'] = 0.0

logDataset['TPS'] = 0.0



# Remove unneeded dates in the index dataset

dateList = []

# Checks for dates that are not in the ethereum datasets

for i in crixDataset.index:

    if (not(crixDataset['date'].at[i] >= pd.Timestamp('2015-08-31') and
crixDataset['date'].at[i] <= pd.Timestamp(

            '2018-02-20'))):

        dateList.append(crixDataset['date'].index[i])



# Delete the rows with those dates to make the same amount of rows between the crix
dataset and ethereum dataset.

crixDataset = crixDataset.drop(dateList)
```

```
crixDataset = crixDataset.reset_index()

crixDataset = crixDataset.drop(columns='index')



# Go through the cleaned up dataset and use the price column to use as
CRIX_crypto_index.

for i in dataset.index:

    dataset['CRIX_crypto_index'].at[i] = crixDataset['price'].at[i]

    logDataset['CRIX_crypto_index'].at[i] = crixDataset['price'].at[i]

# Apply log 10 to the log ethereum dataset.

logDataset['CRIX_crypto_index'] = np.log10(logDataset['CRIX_crypto_index'])



#Creating valuations metrics and ratios for the dataset-formual

for i in dataset.index:

    dataset['sales'].at[i] = (dataset['eth_gasprice'].at[i] * 0.00000002 *
dataset['eth_etherprice'].at[i])

    dataset['cost'].at[i] = (3 * dataset['eth_blocks'].at[i] * dataset['eth_etherprice'].at[i])

    dataset['earnings'].at[i] = (dataset['sales'].at[i] - dataset['cost'].at[i])

    dataset['EPS'].at[i] = (dataset['earnings'].at[i] / dataset['eth_ethersupply'].at[i])

    dataset['PE_ratio'].at[i] = (dataset['eth_marketcap'].at[i] / dataset['eth_tx'].at[i])

    dataset['price_per_sales'].at[i] = dataset['eth_etherprice'].at[i] / dataset['sales'].at[i]
```

```
    dataset['gasprice_per_trans'].at[i] = (dataset['eth_gasprice'].at[i] * 0.00000002 *
dataset['eth_etherprice']).at[i] / dataset['eth_tx'].at[i]

    dataset['gasprice_per_blocks'].at[i] = ((dataset['eth_gasprice'].at[i]  * 0.00000002 *
dataset['eth_etherprice'].at[i]) / dataset['eth_blocks'].at[i])

    dataset['TPS'].at[i] = (dataset['eth_tx'].at[i] / 84600.0)

    dataset['hashrate_per_TPD'].at[i] = (dataset['eth_hashrate'].at[i] /
dataset['eth_tx'].at[i])

    dataset['blocksize_per_TPD'].at[i] = (dataset['eth_blocksize'].at[i] /
dataset['eth_tx'].at[i])



#Creating valuations metrics and ratios for the log-dataset

for i in logDataset.index:

    logDataset['sales'].at[i] = (logDataset['eth_gasprice'].at[i] * 0.00000002 *
logDataset['log10_eth_etherprice'].at[i])

    logDataset['cost'].at[i] = (3 * logDataset['log10_eth_blocks'].at[i] *
logDataset['log10_eth_etherprice'].at[i])

    logDataset['earnings'].at[i] = (logDataset['sales'].at[i] - logDataset['cost'].at[i])

    logDataset['EPS'].at[i] = (logDataset['earnings'].at[i] /
logDataset['log10_eth_ethersupply'].at[i])

    logDataset['PE_ratio'].at[i] = (logDataset['log10_eth_marketcap'].at[i] /
logDataset['eth_tx'].at[i])

    if (logDataset['log10_eth_ethersupply'].at[i] == 0 or logDataset['sales'].at[i] == 0):

        logDataset['price_per_sales'].at[i] = 0

    else:
```

```
    logDataset['price_per_sales'].at[i] = (logDataset['log10_eth_etherprice'].at[i] /
logDataset['sales'].at[i])

  logDataset['gasprice_per_trans'].at[i] = ((logDataset['eth_gasprice'].at[i] *
0.00000002 * logDataset['log10_eth_etherprice'].at[i]) / logDataset['eth_tx'].at[i])

  logDataset['gasprice_per_blocks'].at[i] = ((logDataset['eth_gasprice'].at[i] *
0.00000002 * logDataset['log10_eth_etherprice'].at[i]) /
logDataset['log10_eth_blocks'].at[i])

  logDataset['TPS'].at[i] = (logDataset['eth_tx'].at[i] / 84600.0)

  logDataset['hashrate_per_TPD'].at[i] = (logDataset['log10_eth_hashrate'].at[i] /
logDataset['eth_tx'].at[i])

  logDataset['blocksize_per_TPD'].at[i] = (logDataset['log10_eth_blocksize'].at[i] /
logDataset['eth_tx'].at[i])



# Dataset

print (dataset)

dataset.describe()



# Log Dataset

print (logDataset)

logDataset.describe()



# Histogram of variables of the features of the network in order to visualize how the
data is distributed.

sns.distplot(dataset['eth_etherprice'])
```

```
sns.distplot(dataset['eth_tx'])

sns.distplot(dataset['eth_supply'])

sns.distplot(dataset['eth_marketcap'])

sns.distplot(dataset['eth_ethersupply'])

sns.distplot(dataset['eth_hashrate'])

sns.distplot(dataset['eth_blocks'])

sns.distplot(dataset['eth_difficulty'])

sns.distplot(dataset['eth_uncles'])

sns.distplot(dataset['eth_blocksize'])

sns.distplot(dataset['eth_gasprice'])

sns.distplot(dataset['eth_gaslimit'])

sns.distplot(dataset['eth_gasused'])


#Histogram of transformed variables to log 10

sns.distplot(logDataset['log10_eth_etherprice'])

sns.distplot(logDataset['eth_tx'])

sns.distplot(logDataset['eth_supply'])

sns.distplot(logDataset['log10_eth_hashrate'])

sns.distplot(logDataset['log10_eth_blocksize'])
```

```
#Histogram of the valuations metrics/ratios and CRIX Index

sns.distplot(dataset['CRIX_crypto_index'])

sns.distplot(dataset['sales'])

sns.distplot(dataset['cost'])

sns.distplot(dataset['earnings'])

sns.distplot(dataset['EPS'])

sns.distplot(dataset['PE_ratio'])

sns.distplot(dataset['price_per_sales'])

sns.distplot(dataset['gasprice_per_trans'])

sns.distplot(dataset['gasprice_per_blocks'])

sns.distplot(dataset['hashrate_per_TPD'])

sns.distplot(dataset['blocksize_per_TPD'])

sns.distplot(dataset['TPS'])


#Histogram of valuations metrics/ratios transformed to log 10

sns.distplot(logDataset['sales'])

sns.distplot(logDataset['cost'])

sns.distplot(logDataset['EPS'])

sns.distplot(logDataset['PE_ratio'])
```

```
# Boxplot of the feature of the Network

plt.boxplot(dataset['eth_etherprice'])

plt.xlabel("eth_etherprice")

plt.boxplot(dataset['eth_tx'])

plt.xlabel("eth_tx")

plt.boxplot(dataset['eth_supply'])

plt.xlabel("eth_supply")

plt.boxplot(dataset['eth_hashrate'])

plt.xlabel("hashrate")

plt.boxplot(dataset['eth_blocksize'])

plt.xlabel("blocksize")

plt.boxplot(dataset['eth_marketcap'])

plt.boxplot(dataset['eth_ethersupply'])

plt.boxplot(dataset['eth_blocks'])

plt.boxplot(dataset['eth_difficulty'])

plt.boxplot(dataset['eth_uncles'])

plt.boxplot(dataset['eth_gasprice'])

plt.boxplot(dataset['eth_gaslimit'])

plt.boxplot(dataset['eth_gasused'])
```

```
# Boxplots of valuations metrics/ratios and Crix Cryto Index

plt.boxplot(dataset['CRIX_crypto_index'])

plt.xlabel("CRIX_crypto_index")

plt.boxplot(dataset['sales'])

plt.xlabel("sales")

plt.boxplot(dataset['cost'])

plt.xlabel("cost")

plt.boxplot(dataset['earnings'])

plt.xlabel("earnings")

plt.boxplot(dataset['EPS'])

plt.xlabel("EPS")

plt.boxplot(dataset['PE_ratio'])

plt.xlabel("PE_ratio")

plt.boxplot(dataset['TPS'])

plt.xlabel("TPS")


 Correlation matrix with the dataset

 correlation = dataset.corr()

 print (correlation)

 plt.title('Heatmap of Correlation Matrix')
```

```
sns.heatmap(correlation,
        xticklabels=correlation.columns.values,
        yticklabels=correlation.columns.values, annot=False, cmap="Blues")
plt.show()


# Correlation matrix with log-Dataset
correlation = logDataset.corr()
print (correlation)
plt.title('Heatmap of Correlation Matrix')
sns.heatmap(correlation,
        xticklabels=correlation.columns.values,
        yticklabels=correlation.columns.values, annot=False, cmap="Blues")
plt.show()


#Scatterplot with price to clearly visualize the relation with independent variables.
# Select lowest correlation variables with price in order to visualize the relationship
more clearly.
sns.pairplot(logDataset,
x_vars=['blocksize_per_TPD','price_per_sales','log10_eth_uncles'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

# Select high correlation variables with price in order to compare with the lowest one.
```

```
sns.pairplot(logDataset, x_vars=['sales','cost','earnings'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

# Select sales (transaction fees), because it is the highest correlation variable in order
to visualize the

# multicollinearity condition with the high correlated variable.

sns.pairplot(logDataset, x_vars=['cost','EPS','gasprice_per_blocks'], y_vars='sales', size
=5, aspect=0.7, kind='reg')

# Select cost (reward to miners) because it is the second highest correlation variable in
order to visualize the

# multicollinearity condition with the high correlated variable.

sns.pairplot(logDataset, x_vars=['EPS','eth_tx','earnings'], y_vars='cost', size =5,
aspect=0.7, kind='reg')



# Scatterplot by independent variables with price to visualize if meet the key
assumption of linear relationship of the linear multiple regression.

sns.pairplot(logDataset, x_vars=['log10_eth_blocks','log10_eth_uncles','eth_gasprice'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset,
x_vars=['log10_eth_hashrate','log10_eth_blocksize','log10_eth_difficulty'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset, x_vars=['eth_tx','eth_gasused','eth_supply'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset, x_vars=['log10_eth_ethersupply','eth_gaslimit'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset, x_vars=['CRIX_crypto_index','sales','cost'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')
```

```
sns.pairplot(logDataset, x_vars=['earnings','EPS','PE_ratio'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset,
x_vars=['price_per_sales','gasprice_per_trans','gasprice_per_blocks'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')

sns.pairplot(logDataset, x_vars=['hashrate_per_TPD','blocksize_per_TPD' ,'TPS'],
y_vars='log10_eth_etherprice', size =5, aspect=0.7, kind='reg')



# Functions for ploting every variable with the logo of Ethereum

# Takes in "dataset['column_name']" argument.

def xPerTimeCurve(colName):

    # Skip the Date column.

    if (colName != 'Date(UTC)'):

        x = dataset['Date(UTC)']

        y = dataset[colName]

        plt.title(colName + " per Time")

        plt.plot(x, y, color='r')

        mdates.DateFormatter('%Y %M')

        plt.xticks(rotation=90)

        plt.show()



def logxPerTimeCurve(colName):
```

```
    # Skip the Date column.

    if (colName != 'Date(UTC)'):

        x = dataset['Date(UTC)']

        y = logDataset[colName]

        plt.title(colName + " per Time")

        plt.plot(x, y, color='b')

        mdates.DateFormatter('%Y %M')

        plt.xticks(rotation=90)

        plt.show()


def logoPlot(col1):

    # Starts off with downloading the ethereum image of the web.

    if sys.version_info[0] < 3:

        import urllib2 as urllib

        eth_img = urllib.urlopen(

"https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Ethereum_logo_2014
.svg/256px-Ethereum_logo_2014.svg.png")

    else:

        import urllib

        eth_img = urllib.request.urlopen(
```

```
"https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Ethereum_logo_2014
.svg/256px-Ethereum_logo_2014.svg.png")

    # Does some formatting to the file.

    image_file = io.BytesIO(eth_img.read())

    eth_im = Image.open(image_file)

    width_eth_im, height_eth_im = eth_im.size

    # Set the size of the image.

    eth_im = eth_im.resize((int(eth_im.size[0] * 0.5), int(eth_im.size[1] * 0.5)),
Image.ANTIALIAS)

    # Settings for matplotlib properties. Set axes to display numbers on all axes

    fig, (ax1, ax2) = plt.subplots(2,1, gridspec_kw = {'height_ratios':[3, 1]})

    ax1.set_ylabel(col1,fontsize=12)

    ax2.set_ylabel('eth_tx',fontsize=12)

    ax2.set_yticks([int('%d000000000'%i) for i in range(10)])

    ax2.set_yticklabels(range(10))

    ax1.set_xticks([datetime.date(i,j,1) for i in range(2013,2019) for j in [1,7]])

    ax1.set_xticklabels('')

    ax2.set_xticks([datetime.date(i,j,1) for i in range(2013,2019) for j in [1,7]])

    ax2.set_xticklabels([datetime.date(i,j,1).strftime('%b %Y')  for i in range(2013,2019)
for j in [1,7]])
```

```
    # Plotting the big "x per time" plot

    ax1.plot(dataset['Date(UTC)'].astype(datetime.datetime), dataset[col1])

    # Plotting the smaller bar plot for eth_tx

    ax2.bar(dataset['Date(UTC)'].astype(datetime.datetime).values, dataset['eth_tx'])


    # Set figure settings for the image location.

    fig.tight_layout()

    fig.figimage(eth_im, 50, 25, zorder=3, alpha=.6)

    plt.title('Time')

    plt.show()


# Change the logo in fig.figmage

# FUNCTIONS END

# Visualise all variables against date with the ETH logo.

for i in dataset.columns:

    logoPlot(i)


############################################## Linear regression with the log-Dataset
```

```
#Identify dependent variable and independent variable

logDataset.describe()

X = logDataset[['eth_tx', 'eth_supply',
'log10_eth_hashrate','log10_eth_difficulty','log10_eth_blocks','log10_eth_uncles','log1
0_eth_blocksize','eth_gasprice','eth_gaslimit','eth_gasused','log10_eth_ethersupply','C
RIX_crypto_index','sales',
'cost','earnings','EPS','PE_ratio','price_per_sales','gasprice_per_trans','gasprice_per_blo
cks','hashrate_per_TPD','blocksize_per_TPD','TPS']]

y = logDataset['log10_eth_etherprice']



#cross-validation with 20% test set

X_train, X_test, Y_train, Y_test = sklearn.cross_validation.train_test_split(X, y,
test_size=.20)



#linear regression

lm = LinearRegression()

lm.fit(X_train, Y_train)

print("lm.intercept_", lm.intercept_)

print("lm.coef_", lm.coef_)

predict = lm.predict(X_test)

predict

#Evaluation metrics for linear regression

print("MEAN SQUARED ERROR: ", metrics.mean_squared_error(Y_test, predict))
```

```
print("R2 SCORE: ", metrics.r2_score(Y_test, predict))

predict.var()



#plotting the coefficient in a table with their respective names

L = pd.DataFrame(list(zip(logDataset[['eth_tx', 'eth_supply',
'log10_eth_hashrate','log10_eth_difficulty','log10_eth_blocks','log10_eth_uncles','log1
0_eth_blocksize','eth_gasprice','eth_gaslimit','eth_gasused','log10_eth_ethersupply','C
RIX_crypto_index','sales',
'cost','earnings','EPS','PE_ratio','price_per_sales','gasprice_per_trans','gasprice_per_blo
cks','hashrate_per_TPD','blocksize_per_TPD','TPS']].columns, lm.coef_)),

        columns=["Independent variables", "Coefficients"])

print (L)



# Plot of residuals

plt.scatter(lm.predict(X_train), lm.predict(X_train) - Y_train, c='b',s=40,alpha=0.5)

plt.scatter(lm.predict(X_test), lm.predict(X_test) - Y_test, c='g',s=40,alpha=0.5)

plt.hlines(y=0, xmin=0, xmax=3)

plt.title("")

plt.ylabel('Residuals')

plt.xlabel('Fitted values')

plt.ylim(ymax=0.005)

plt.ylim(ymin=-0.01)
```

```
plt.show()



# Plot the actual values vs. predicted values of ether

clf = LinearRegression()

clf.fit(logDataset.values[:,1:], logDataset['log10_eth_etherprice'])

predict = clf.predict(logDataset.values[:,1:])

predict = predict.astype(float)

plt.scatter(logDataset['log10_eth_etherprice'], predict , alpha = .5)

plt.plot([0, 5], [0, 5], "--k")

plt.axis("tight")

plt.xlabel("Actual price")

plt.ylabel("Predicted price")

plt.show()



########################################### Linear regression without log 10
(on original dataset)

#Identify dependent variable and independent variable

dataset.describe()

X = dataset[['eth_tx', 'eth_supply',
'eth_hashrate','eth_difficulty','eth_blocks','eth_uncles','eth_blocksize','eth_gasprice','e
th_gaslimit','eth_gasused','eth_ethersupply','CRIX_crypto_index','sales',
```

```
'cost','earnings','EPS','PE_ratio','price_per_sales','gasprice_per_trans','gasprice_per_blo
cks','hashrate_per_TPD','blocksize_per_TPD','TPS']]

y = dataset['eth_etherprice']



#cross-validation with 20% test set

X_train, X_test, Y_train, Y_test = sklearn.cross_validation.train_test_split(X, y,
test_size=.20)



#linear regression

lm = LinearRegression()

lm.fit(X_train, Y_train)

print("lm.intercept_", lm.intercept_)

print("lm.coef_", lm.coef_)

predict = lm.predict(X_test)

predict



#Evaluation metrics for linear regression

print("MEAN SQUARED ERROR: ", metrics.mean_squared_error(Y_test, predict))

print("R2 SCORE: ", metrics.r2_score(Y_test, predict))

predict.var()
```

```
#plotting the coefficient in a table with their respective names

L = pd.DataFrame(list(zip(dataset[['eth_tx', 'eth_supply',
'eth_hashrate','eth_difficulty','eth_blocks','eth_uncles','eth_blocksize','eth_gasprice','e
th_gaslimit','eth_gasused','eth_ethersupply','CRIX_crypto_index','sales',
'cost','earnings','EPS','PE_ratio','price_per_sales','gasprice_per_trans','gasprice_per_blo
cks','hashrate_per_TPD','blocksize_per_TPD','TPS']].columns, lm.coef_)),

        columns=["Independent variables", "Coefficients"])

print (L)



# Plot of residuals

plt.scatter(lm.predict(X_train), lm.predict(X_train) - Y_train, c='b',s=40,alpha=0.5)

plt.scatter(lm.predict(X_test), lm.predict(X_test) - Y_test, c='g',s=40,alpha=0.5)

plt.hlines(y=0, xmin=0, xmax=3)

plt.title("")

plt.ylabel('Residuals')

plt.xlabel('Fitted values')

plt.ylim(ymax=0.005)

plt.ylim(ymin=-0.01)

plt.show()



# Plot the actual values vs. predicted values of ether

clf = LinearRegression()
```

```
clf.fit(logDataset.values[:,1:], logDataset['log10_eth_etherprice'])

predict = clf.predict(logDataset.values[:,1:])

predict = predict.astype(float)

plt.scatter(logDataset['log10_eth_etherprice'], predict , alpha = .5)

plt.plot([0, 5], [0, 5], "--k")

plt.axis("tight")

plt.xlabel("Actual price")

plt.ylabel("Predicted price")

plt.show()



############################################ Linear regression - redefining the linear regression with log dataset to identify the best trade-off # of variables and MSE to achieve the proposed model

#Identify dependent variable and independent variable

logDataset.describe()

# To identify the best-trade-off of # variables and MSE

X = logDataset[['gasprice_per_blocks']]

X = logDataset[['gasprice_per_blocks', 'sales']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings', 'gasprice_per_trans']]
```

```
X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings', 'gasprice_per_trans',
'PE_ratio']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings', 'gasprice_per_trans',
'PE_ratio', 'blocksize_per_TPD']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings', 'gasprice_per_trans',
'PE_ratio', 'blocksize_per_TPD', 'hashrate_per_TPD']]

X = logDataset[['gasprice_per_blocks', 'sales', 'cost', 'earnings', 'gasprice_per_trans',
'PE_ratio', 'blocksize_per_TPD', 'hashrate_per_TPD', 'eth_supply']]



#Proposed model

# Cut off - best trade-off - First 3 top variables

X = logDataset[['gasprice_per_blocks', 'sales', 'cost']]

y = logDataset['log10_eth_etherprice']



#Cross-validation

X_train, X_test, Y_train, Y_test = sklearn.cross_validation.train_test_split(X, y,
test_size=.20)



#linear regression

lm = LinearRegression()

lm.fit(X_train, Y_train)

print("lm.intercept_", lm.intercept_)
```

```
print("lm.coef_", lm.coef_)

predict = lm.predict(X_test)

predict


#Evaluation metrics for linear regression

print("MEAN SQUARED ERROR: ", metrics.mean_squared_error(Y_test, predict))

print("R2 SCORE: ", metrics.r2_score(Y_test, predict))

predict.var()


#plotting the coefficient in a table with their respective names

L = pd.DataFrame(list(zip(logDataset[['gasprice_per_blocks', 'sales', 'cost']].columns,
lm.coef_)),

          columns=["Independent variables", "Coefficients"])

print (L)



# Plot of residuals

plt.scatter(lm.predict(X_train), lm.predict(X_train) - Y_train, c='b',s=40,alpha=0.5)

plt.scatter(lm.predict(X_test), lm.predict(X_test) - Y_test, c='g',s=40,alpha=0.5)

plt.hlines(y=0, xmin=0, xmax=3)
```

```
plt.title("")

plt.ylabel('Residuals')

plt.xlabel('Fitted values')

plt.ylim(ymax=0.005)

plt.ylim(ymin=-0.01)

plt.show()


# Plot the actual values vs. predicted values of ether

clf = LinearRegression()

clf.fit(logDataset.values[:,1:], logDataset['log10_eth_etherprice'])

predict = clf.predict(logDataset.values[:,1:])

predict = predict.astype(float)

plt.scatter(logDataset['log10_eth_etherprice'], predict , alpha = .5)

plt.plot([0, 5], [0, 5], "--k")

plt.axis("tight")

plt.xlabel("Actual price")

plt.ylabel("Predicted price")

plt.show()
```

```
##################################### Changed the formula of the variable of Sales (transaction fees) to the real one, which include gas limit in order to analyze the value of ether with figures



#Creating valuations metrics and ratios for the dataset

for i in dataset.index:

   dataset['sales'].at[i] = (dataset['eth_gasprice'].at[i] * dataset['eth_gaslimit'].at[i] * 0.00000002 * dataset['eth_etherprice'].at[i])

   dataset['cost'].at[i] = (3 * dataset['eth_blocks'].at[i] * dataset['eth_etherprice'].at[i])

   dataset['earnings'].at[i] = (dataset['sales'].at[i] - dataset['cost'].at[i])

   dataset['EPS'].at[i] = (dataset['earnings'].at[i] / dataset['eth_ethersupply'].at[i])

   dataset['PE_ratio'].at[i] = (dataset['eth_marketcap'].at[i] / dataset['eth_tx'].at[i])

   dataset['price_per_sales'].at[i] = dataset['eth_etherprice'].at[i] / dataset['sales'].at[i]

   dataset['gasprice_per_trans'].at[i] = (dataset['eth_gasprice'].at[i] * 0.00000002 * dataset['eth_etherprice']).at[i] / dataset['eth_tx'].at[i]

   dataset['gasprice_per_blocks'].at[i] = ((dataset['eth_gasprice'].at[i]  * 0.00000002 * dataset['eth_etherprice'].at[i]) / dataset['eth_blocks'].at[i])

   dataset['TPS'].at[i] = (dataset['eth_tx'].at[i] / 84600.0)

   dataset['hashrate_per_TPD'].at[i] = (dataset['eth_hashrate'].at[i] / dataset['eth_tx'].at[i])

   dataset['blocksize_per_TPD'].at[i] = (dataset['eth_blocksize'].at[i] / dataset['eth_tx'].at[i])

   # dataset['sharp_ratio'].at[i] = (dataset['returns_qrt'].at[i] / dataset['std_dev_qrt'].at[i])
```

```
#Creating valuations metrics and ratios for the log-dataset

for i in logDataset.index:

    logDataset['sales'].at[i] = (logDataset['eth_gasprice'].at[i] *
logDataset['eth_gaslimit'].at[i] * 0.00000002 * logDataset['log10_eth_etherprice'].at[i])

    logDataset['cost'].at[i] = (3 * logDataset['log10_eth_blocks'].at[i] *
logDataset['log10_eth_etherprice'].at[i])

    logDataset['earnings'].at[i] = (logDataset['sales'].at[i] - logDataset['cost'].at[i])

    logDataset['EPS'].at[i] = (logDataset['earnings'].at[i] /
logDataset['log10_eth_ethersupply'].at[i])

    logDataset['PE_ratio'].at[i] = (logDataset['log10_eth_marketcap'].at[i] /
logDataset['eth_tx'].at[i])

    if (logDataset['log10_eth_ethersupply'].at[i] == 0 or logDataset['sales'].at[i] == 0):

        logDataset['price_per_sales'].at[i] = 0

    else:

        logDataset['price_per_sales'].at[i] = (logDataset['log10_eth_etherprice'].at[i] /
logDataset['sales'].at[i])

    logDataset['gasprice_per_trans'].at[i] = ((logDataset['eth_gasprice'].at[i] *
0.00000002 * logDataset['log10_eth_etherprice'].at[i]) / logDataset['eth_tx'].at[i])

    logDataset['gasprice_per_blocks'].at[i] = ((logDataset['eth_gasprice'].at[i] *
0.00000002 * logDataset['log10_eth_etherprice'].at[i]) /
logDataset['log10_eth_blocks'].at[i])

    logDataset['TPS'].at[i] = (logDataset['eth_tx'].at[i] / 84600.0)
```

```
    logDataset['hashrate_per_TPD'].at[i] = (logDataset['log10_eth_hashrate'].at[i] /
logDataset['eth_tx'].at[i])

    logDataset['blocksize_per_TPD'].at[i] = (logDataset['log10_eth_blocksize'].at[i] /
logDataset['eth_tx'].at[i])



#dataset

print (dataset)

dataset.describe()



#log dataset

print (logDataset)

logDataset.describe()
```