Advanced Operating Systems Project
CryptAnalytics

Delton M Antony, 18MCMI05
Mtech Artificial Intelligence

ABSTRACT

There are well over a thousand cryptocurrencies in existence today. Investing in cryptocurrencies can lead to substantial profits or crippling losses for the investor. Out of these currencies, which one is the best to invest in? What are the trends in the major cryptocurrencies that can be made visible to the investor? These are the questions that will be answered in the project by using a dataset that logged the major cryptocurrencies throughout these last few years. After the descriptive data analysis, we will try to predict the future values of the major cryptocurrencies using predictive analytics techniques using machine learning to find out if it is possible to make the most optimal investment. This project will try to check if machine learning can predict the future of the ever unpredictable blockchain.

CONTENTS

1. INTRODUCTION

In this project I apply descriptive and predictive analytics in the cryptocurrency dataset to identify which cryptocurrency is the best in terms of return of investment. We take three of the major currencies to perform the analysis and choose the best one that is ideal for investment. We perform exploratory data analysis first. Then we move on to predictive analysis by trying to predict the price of the selected cryptocurrencies.

We are using python3 matplotlib.pyplot and seaborn for exploratory data analysis. We then do the predictions using the scikit learn library.

Tech Stack: ipython, anaconda, jupyter, sklearn, spyder

2. PROBLEM DEFINITION AND DATA-MINING TASK IDENTIFICATION

We have acquired a dataset with the opening, closing, high and low cryptocurrency values for the day along with market-cap and close ratio. The output variable is not defined. We need to define the output variable as the value of the cryptocurrency 30 days after the current day. Our task is to perform exploratory data analysis on the existing data and then predict the values of the cryptocurrencies for 30 days in the future. Since our output variable is a continuous value, this problem can be identified as a regression problem.

3. METHODS AND TECHNIQUES APPLIED

3.1. Data Partitioning:

This is the process of splitting the data into training set and test set. This has to be done in such a way that the split ratio is fair. The split ratio here is 4:1. The scikitlearn model selection package provides the function train_test_split to do this.

3.2. Random Forest Regression:

The Random Forest is one of the most effective machine learning models for predictive analytics, making it an industrial workhorse for machine learning.

The random forest model is a type of additive model that makes predictions by combining decisions from a sequence of base models. More formally we can write this class of models as:

$g(x)=f0(x)+f1(x)+f2(x)+...$

Where the final model g is the sum of simple base models fi. Here, each base classifier is a simple decision tree. This broad technique of using multiple models to obtain better predictive performance is called model ensembling. In random forests, all the base models are constructed independently using a different subsample of the data.

3.3 Gradient Boosting Regression:

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

4. DATASET DESCRIPTION

The dataset contains a total of 7,02,166 records which are all completely filled without any missing or null values. It is of the format csv - comma separated values where each of the records have the attributes separated by commas. The header, which mentions the attribute name for each columns is present by default in the dataset. These 12 columns correspond to the attributes of the cryptocurrency. There is no target column. There both numeric and categorical values in the dataset. We will handle the categorical values by filtering each of them out separately rather than doing one hot encoding. Loading the dataset into the memory takes a comparatively long time.

5. EXPLORATORY DATA ANALYSIS

The following is what the info method, which is a method to describe the dataset in terms of Attribute Names and data type, returned after loading the dataset:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 702166 entries, 2013-04-28 to 2018-01-06
Data columns (total 12 columns):
slug          702166 non-null object
symbol          702166 non-null object
name           702166 non-null object
ranknow         702166 non-null int64
open           702166 non-null float64
high          702166 non-null float64
low           702166 non-null float64
close          702166 non-null float64
volume          702166 non-null int64
market          702166 non-null int64
close_ratio    702166 non-null float64
spread          702166 non-null float64
dtypes: float64(6), int64(3), object(3)
memory usage: 69.6+ MB
```

This data file is for format .csv and is 70 megabytes in size. It contains the info about the three cryptocurrencies along with others like Ripple, Ethereum classic, ZCash etc. We are only concerned with Bitcoin, Ethereum and Litecoin. Hence I filtered them out separately. This reduced the size significantly as follows:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4452 entries, 2013-04-28 to 2018-02-21
Data columns (total 12 columns):
slug          4452 non-null object
symbol          4452 non-null object
name           4452 non-null object
ranknow         4452 non-null int64
open           4452 non-null float64
high          4452 non-null float64
low           4452 non-null float64
close          4452 non-null float64
volume          4452 non-null int64
market          4452 non-null int64
close_ratio    4452 non-null float64
```

```
spread         4452 non-null float64
dtypes: float64(6), int64(3), object(3)
memory usage: 452.2+ KB
```

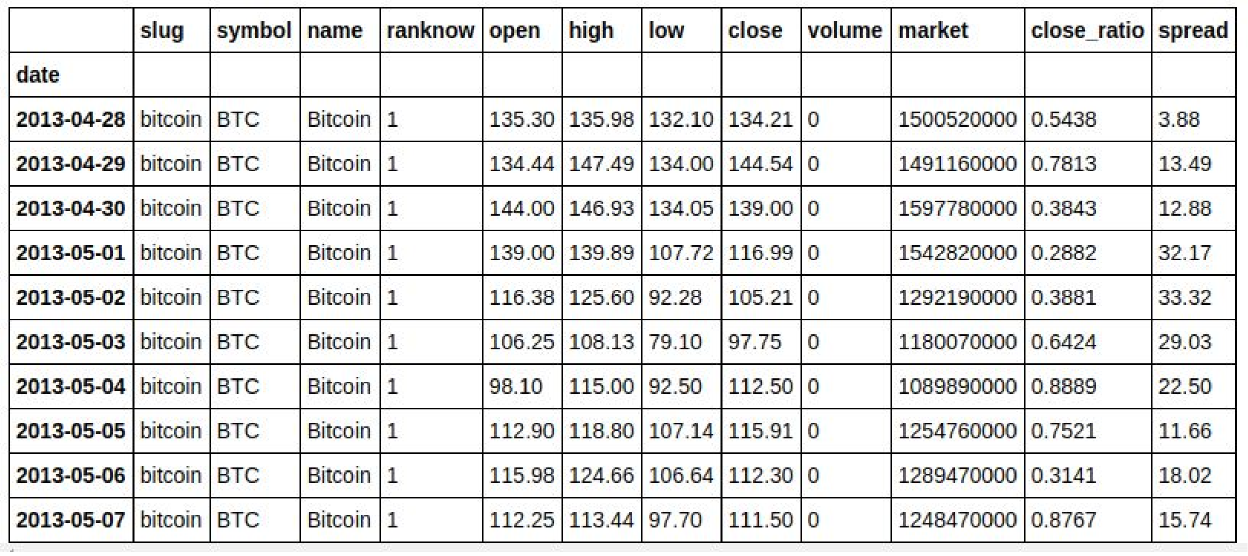
Out of 702166 records, we got the 4452 records that contain information about Bitcoin, Ethereum and Litecoin. Let us see how many records exist for each.

```
Litecoin    1761
Bitcoin     1761
Ethereum     930
Name: name, dtype: int64
```

There are both continuous as well as categorical variables in this dataset. Continuous variables are those that can take a real number value and categorical variables are those that are only able to take a value that belongs to a set of predetermined values.

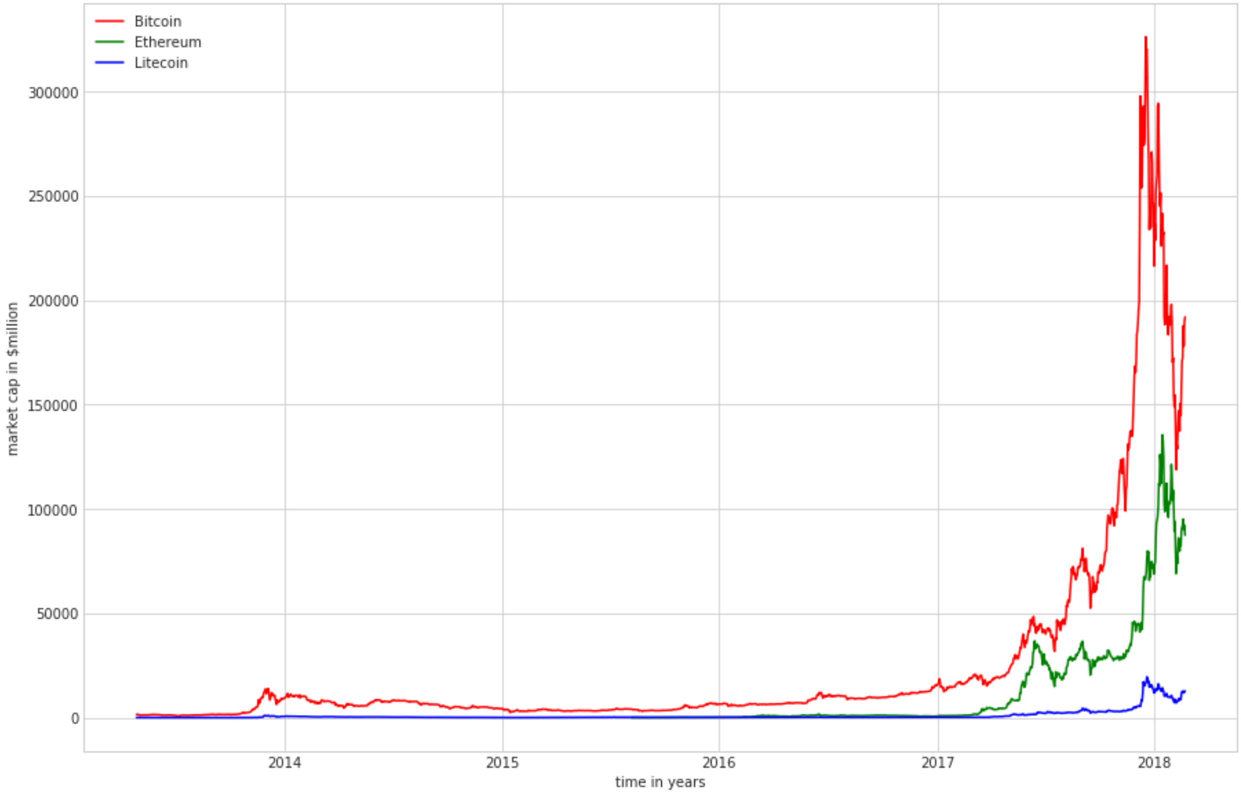
We are visualizing the data using histograms. The following graphs are plotted using the seaborn and matplotlib.pyplot libraries of python3.

The datahead looks like this



	slug	symbol	name	ranknow	open	high	low	close	volume	market	close_ratio	spread
date												
2013-04-28	bitcoin	BTC	Bitcoin	1	135.30	135.98	132.10	134.21	0	1500520000	0.5438	3.88
2013-04-29	bitcoin	BTC	Bitcoin	1	134.44	147.49	134.00	144.54	0	1491160000	0.7813	13.49
2013-04-30	bitcoin	BTC	Bitcoin	1	144.00	146.93	134.05	139.00	0	1597780000	0.3843	12.88
2013-05-01	bitcoin	BTC	Bitcoin	1	139.00	139.89	107.72	116.99	0	1542820000	0.2882	32.17
2013-05-02	bitcoin	BTC	Bitcoin	1	116.38	125.60	92.28	105.21	0	1292190000	0.3881	33.32
2013-05-03	bitcoin	BTC	Bitcoin	1	106.25	108.13	79.10	97.75	0	1180070000	0.6424	29.03
2013-05-04	bitcoin	BTC	Bitcoin	1	98.10	115.00	92.50	112.50	0	1089890000	0.8889	22.50
2013-05-05	bitcoin	BTC	Bitcoin	1	112.90	118.80	107.14	115.91	0	1254760000	0.7521	11.66
2013-05-06	bitcoin	BTC	Bitcoin	1	115.98	124.66	106.64	112.30	0	1289470000	0.3141	18.02
2013-05-07	bitcoin	BTC	Bitcoin	1	112.25	113.44	97.70	111.50	0	1248470000	0.8767	15.74

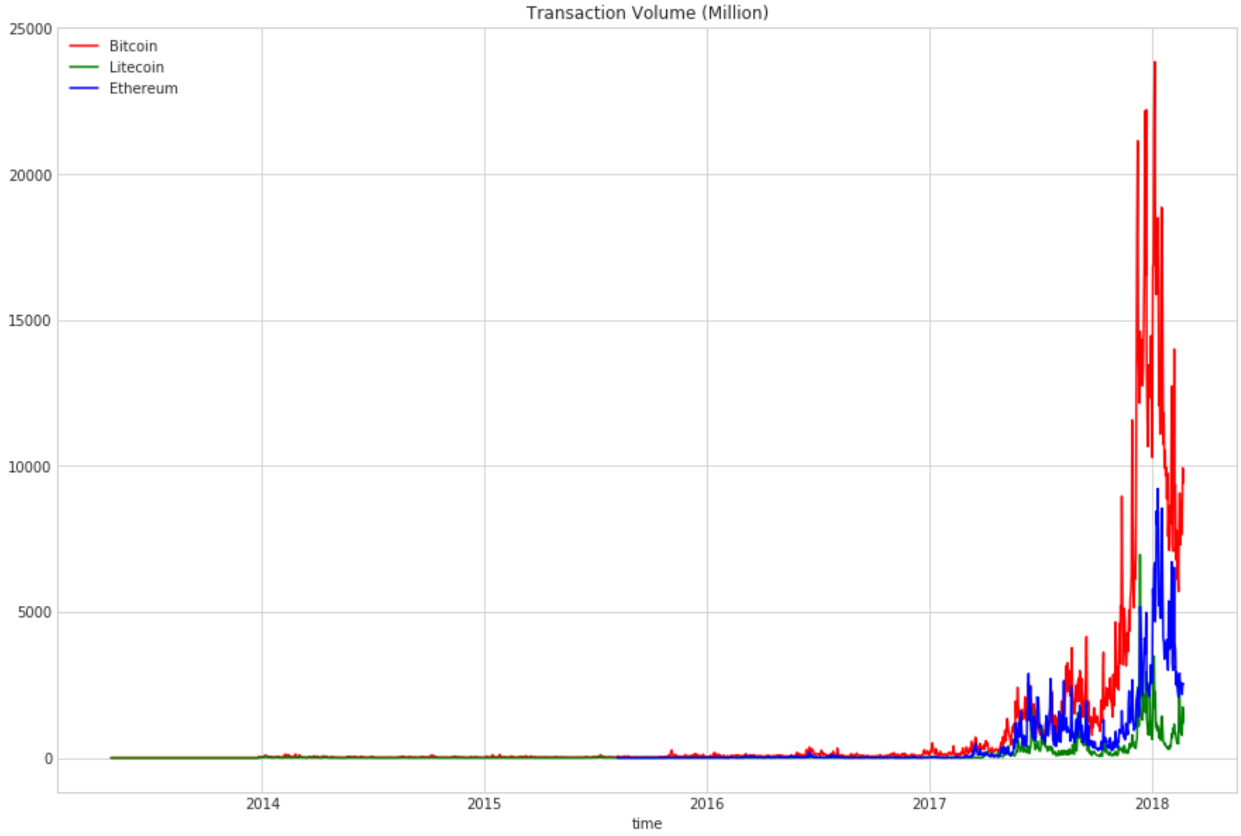
Plotting the market cap of the three currencies:



One important thing I noticed is that a direct correlation can be seen forming between the data. Cryptocurrencies became all the rage after 2017.

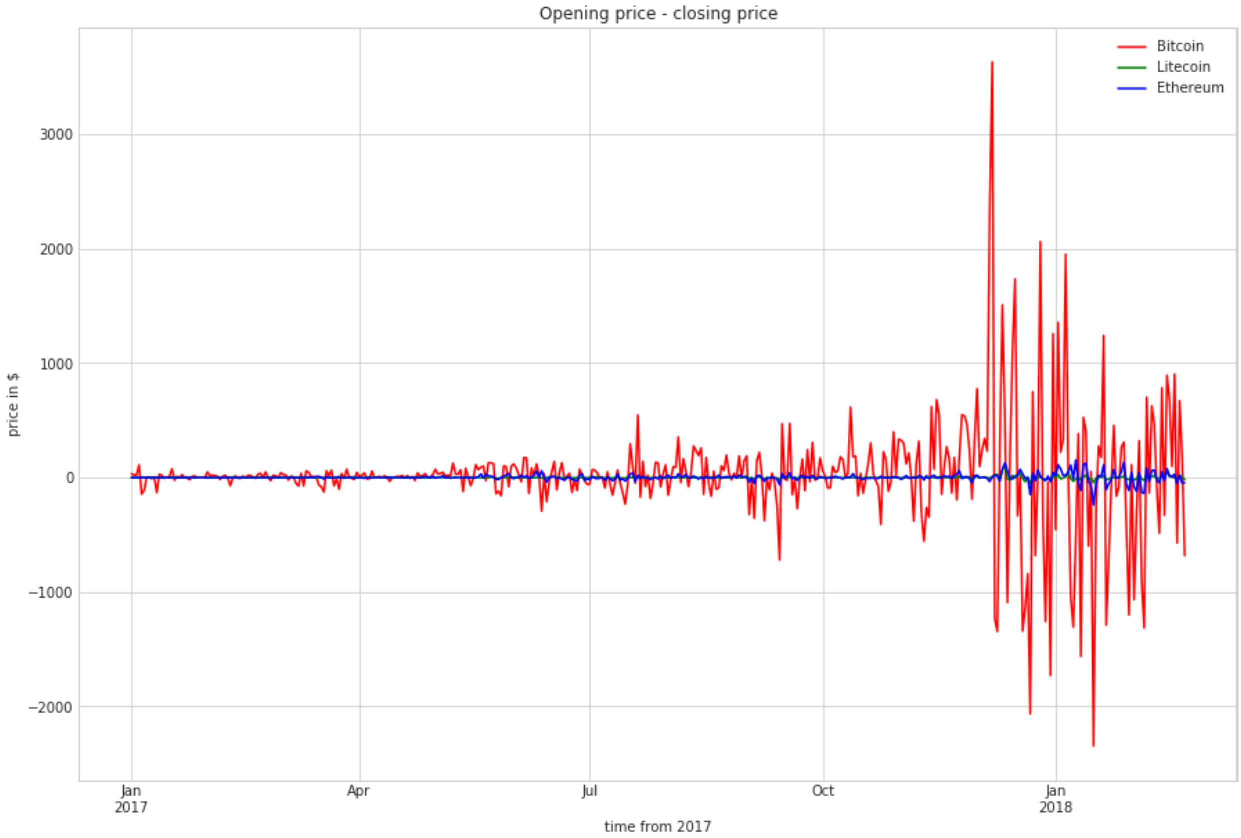
Transaction Volume: One direct result of the number of people that participate in the blockchain is that the value of the cryptocurrency is directly dependent on that. Hence it is important to include Transaction Volume in the data as it can directly influence the price of that currency.

Let us plot it as a function of time.



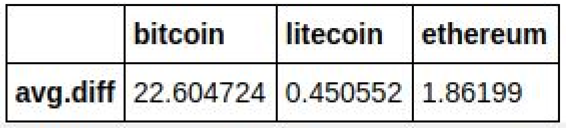
It is evident that nothing much happened in this area until 2017. Before that people showed some interest in bitcoin, but that was it. So, let us only consider those records that were made from 2017.

The difference between opening and closing price of the currencies are plotted as below.



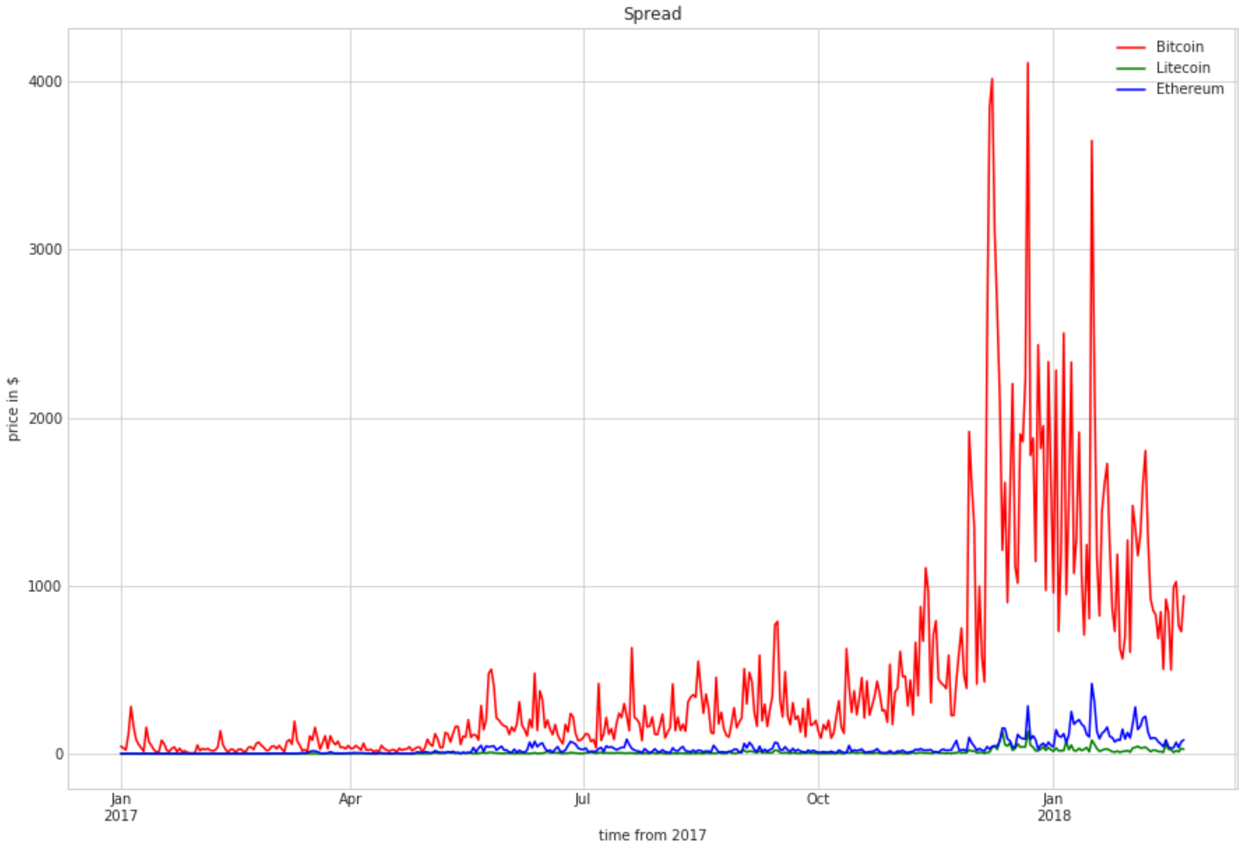
We can see that the difference is the highest in Bitcoin while the other currencies are stable. This may be because the value of those currencies are also proportionately less.

The average difference can be summarized as below.



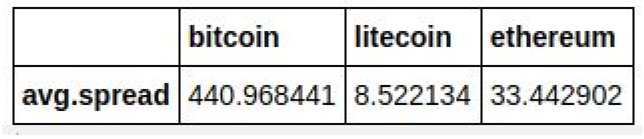
	bitcoin	litecoin	ethereum
avg.diff	22.604724	0.450552	1.86199

Difference between the daily highest and the daily lowest price is given by the feature "spread". Let us plot it with respect to time.



The prices difference from high to low, as seen in the case of opening and closing, is also high in Bitcoin. A similar trend is followed closely by Ethereum and Litecoin.

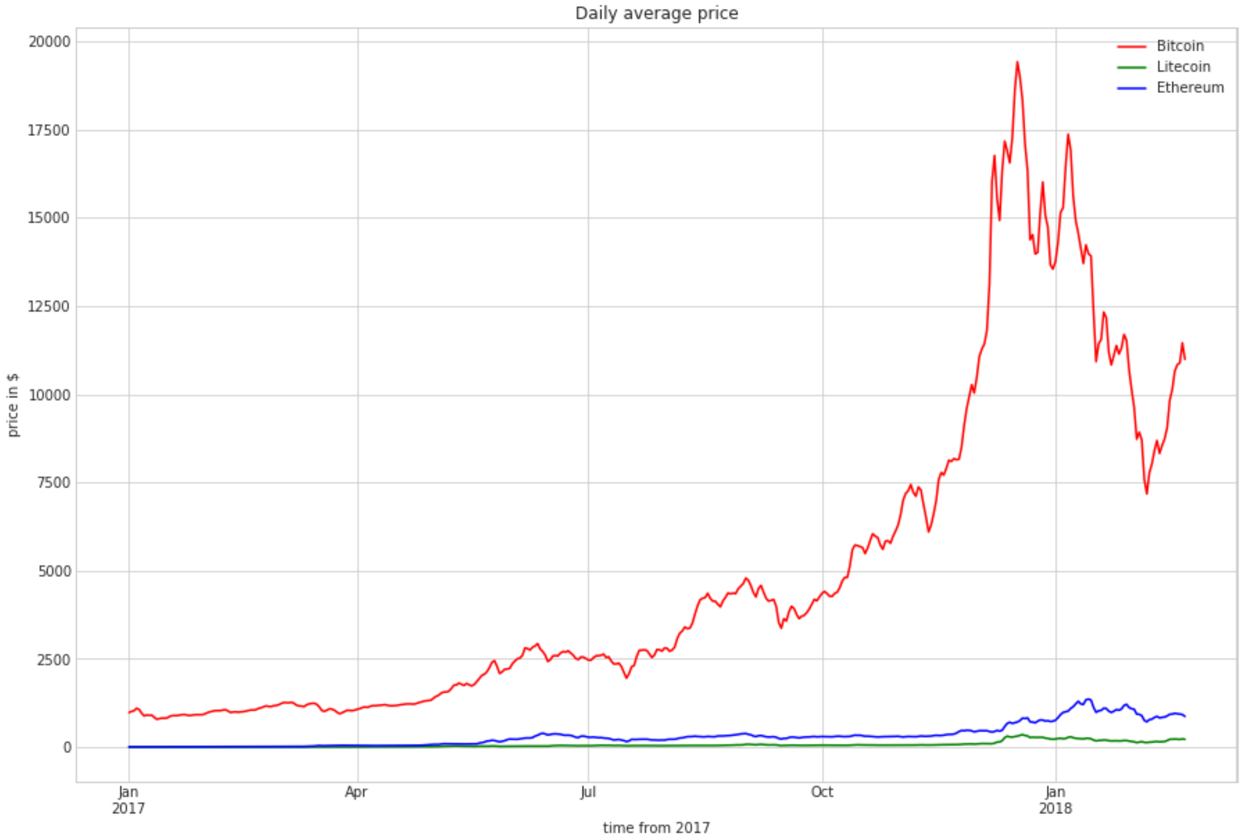
Let us summarize the average spread.



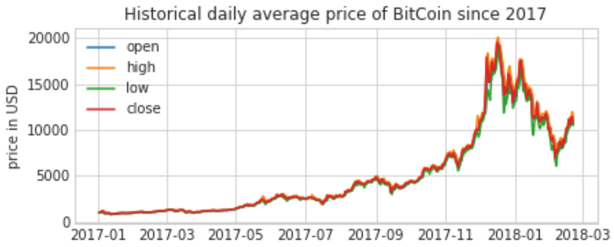
	bitcoin	litecoin	ethereum
avg.spread	440.968441	8.522134	33.442902

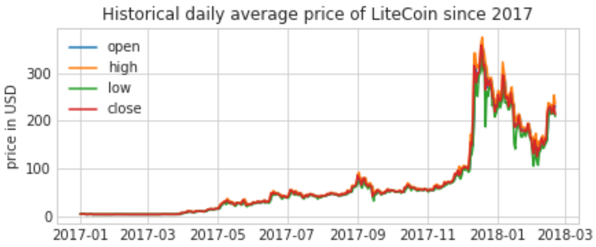
Lets make a daily average price column for the dataset. Here, I define average as (open+close+high+low)/4

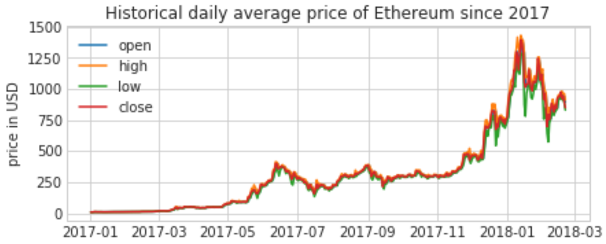
Now let us plot it for the three currencies.



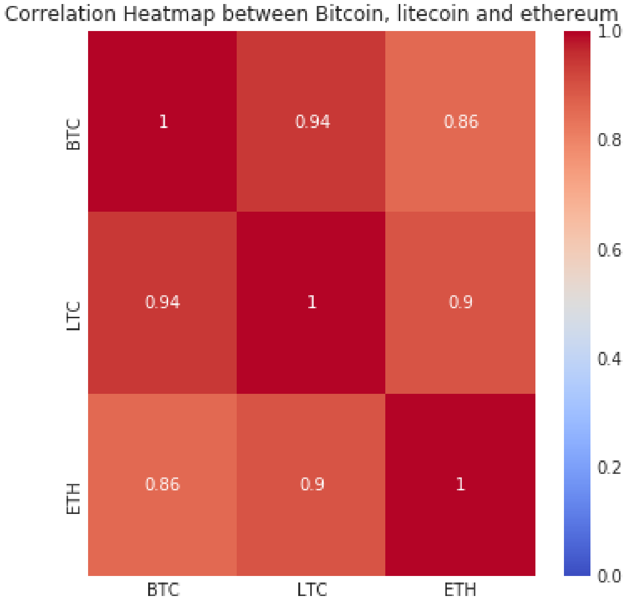
Now let’s plot individually





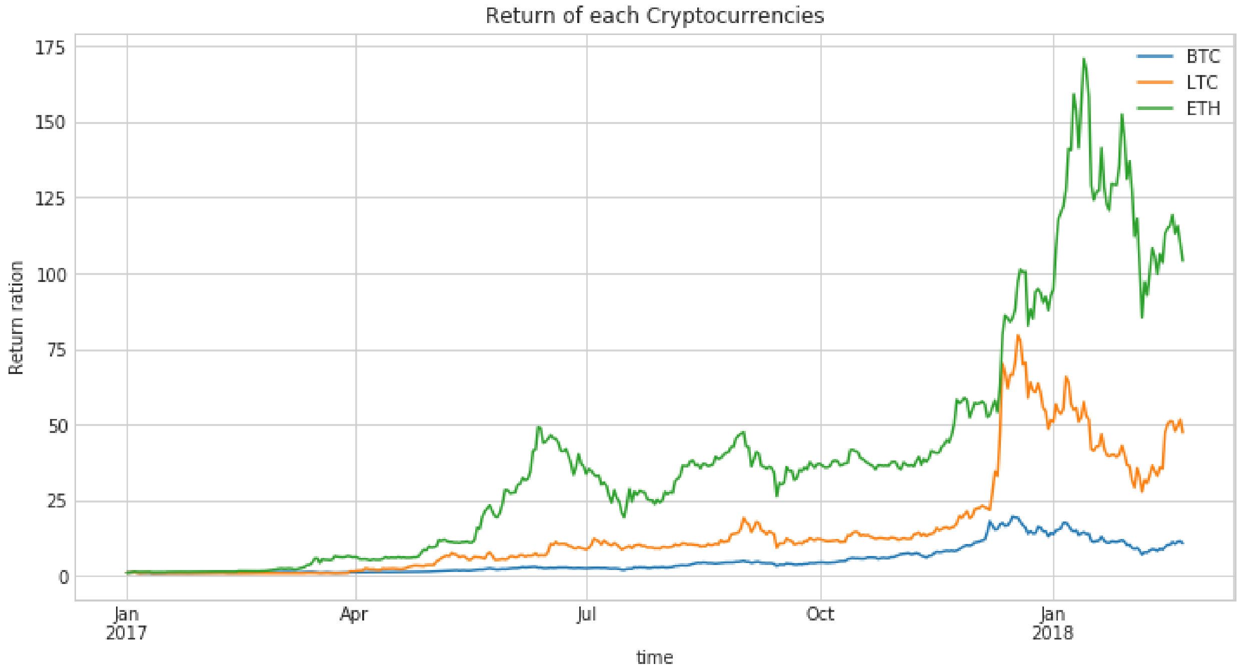


What we can infer from the above is that the trend is similar in shape, but not in scale. The pattern of bitcoin is closely followed by ethereum and litecoin in a much smaller scale ie price.
This hypothesis can be verified by using a correlation matrix. If there is a high positive correlation, then we can conclude that bitcoin price influences the other cryptocurrencies.



We can see that there is a very high positive correlation. This proves the above hypothesis.

Now the most important factor ie the Returns Ratio. Let us see for the three cryptocurrencies.



From this, you can see that Bitcoin became fairly stable since 2017 while Ethereum and Litecoin gives the highest return ratio. From this, it is seen that investing in Ethereum can have a good ROI in the long term despite the frequent fluctuations and casper.

6. DATA PREPROCESSING

Data Preprocessing is the most important part of the CRISP-DM methodology which we are following here. Here, in this case, we need to correct check and impute the missing values, handle categorical data, perform feature selection and finally do the data partitioning for hold out method.

6.1. Checking for null values

The cryptocurrency dataset is near perfect with absolutely no null values as the dataset info method returned 7,02,166 non null objects for all variables.

6.2. Creating a target variable:

At first glance, the problem looks like we need to predict in the records field for the thirty days after the latest record. Here we are doing a trick in order to make a target variable. I will move all daily_avg values thirty lines up. With this, I introduce a one month lag. Then I define a new column - daily_avg_After_Month which will be the dependent variable. Data partitioning is done in 4:1 ratio. Create X_forecast using all models.

6.3. Handling Categorical Variables:

Categorical variables are those variables that can only take values from a finite set of values called its domain. Here, all of the categorical variables except Rank are nominal as opposed to ordinal. We are not performing one hot encoding or get dummies here. Instead, we are again splitting the dataset according to category. One for each cryptocurrency.

6.4. Feature Selection:

Here, we are discarding the unimportant features. A feature can be considered as unimportant if it either does not contribute any meaningful data to the classifier so that the classifier can associate it with the target or if the contribution is so less that the added complexity is a worse trade-off which can be sacrificed without much loss in accuracy. We are dropping features like rank, symbol, etc from the prediction context.

6.5. Data Partitioning:

Data partitioning means how we are splitting the data into training set and testing set. We are doing both the regular hold out method with the sklearn train_test_split. In hold out, our split ratio is 4:1 with a fixed seed so that we get the same split on each trial.

7. RESULTS AND DISCUSSIONS

The results of this project can be divided into two parts - the results of exploratory data analysis and their discussion and the results of data modeling and their discussion.

Let’s start with the discussion of the results of EDA which was given in the end of EDA.

We performed EDA and found certain trends in the data. We can say that in order to get the maximum return of investment, it is advisable to invest in ethereum. From the return of investment graph, ethereum promises to be the best option to invest.

Now, with EDA done, let’s move on to the results we got in data modeling.

The regression models we built here for this are:

1. Random forest Regression
2. Gradient Boosting Regression

For selecting the hyperparameters, we used grid search in python. Grid search is a brute force based technique where you initialize a list with all the parameter values you want to try out for your model instances. Then you pass that to the function that calls the models with different permutations of all the parameters you chose. Note that you should keep the seed or random_state as a constant value. It finds a tuple of optimal parameters for you to fit on the model. Grid search can be considered as a kind of methodical trial and error.

The models used and their results are listed below

For Bitcoin:

Random Forest Regressor
R2: 0.95
MAE: 644.18
MSE: 1061069.81

Gradient Boosting Regressor
R2: 0.94
MAE: 742.00
MSE: 1269684.59

For Ethereum:

Random Forest Regressor
R2: -2.90
MAE: 546.96
MSE: 351518.19

Gradient Boosting Regressor
R2: -2.21
MAE: 489.88
MSE: 290049.90

For Litecoin:

Random Forest Regressor
R2: 0.75
MAE: 16.69
MSE: 1425.96

Gradient Boosting Regressor
R2: 0.62
MAE: 20.82
MSE: 2125.01

Note that the errors seem big as we have not done any feature scaling on the data as we are only using Random Forest Regressor and Gradient Boosting which does not necessarily need scaled data.

8. CONCLUSIONS

I was able to draw out inferences from the data by performing Exploratory Data Analysis. It showed us how the prices of the other currencies are directly correlated to the price of Bitcoin. It seems that Bitcoin was the currency that started the trend of cryptocurrency mining. However, when it comes to return of investment now, investing in Ethereum is a better option for the long term. Note that these results can fluctuate with time as it is the nature of blockchain to do so.

When it comes to predictive analytics, I was able to predict the price movement of Bitcoin, Ethereum and Litecoin for the next 30 unseen days. However, one thing to be noted here is that this model has to be updated every single day as the model's behavior directly corresponds to the data available till the last day. This is the shortcoming of the model. If there is an anomaly in the price of any of the cryptocurrencies one day, the entire model and its predictions will change. This is the reason we will not be able to predict the price of any cryptocurrencies for long term investment. However, if one is able to find out the correct input features, then it might be possible. Till date no one knows the complete list of the features that affect cryptocurrency prices.

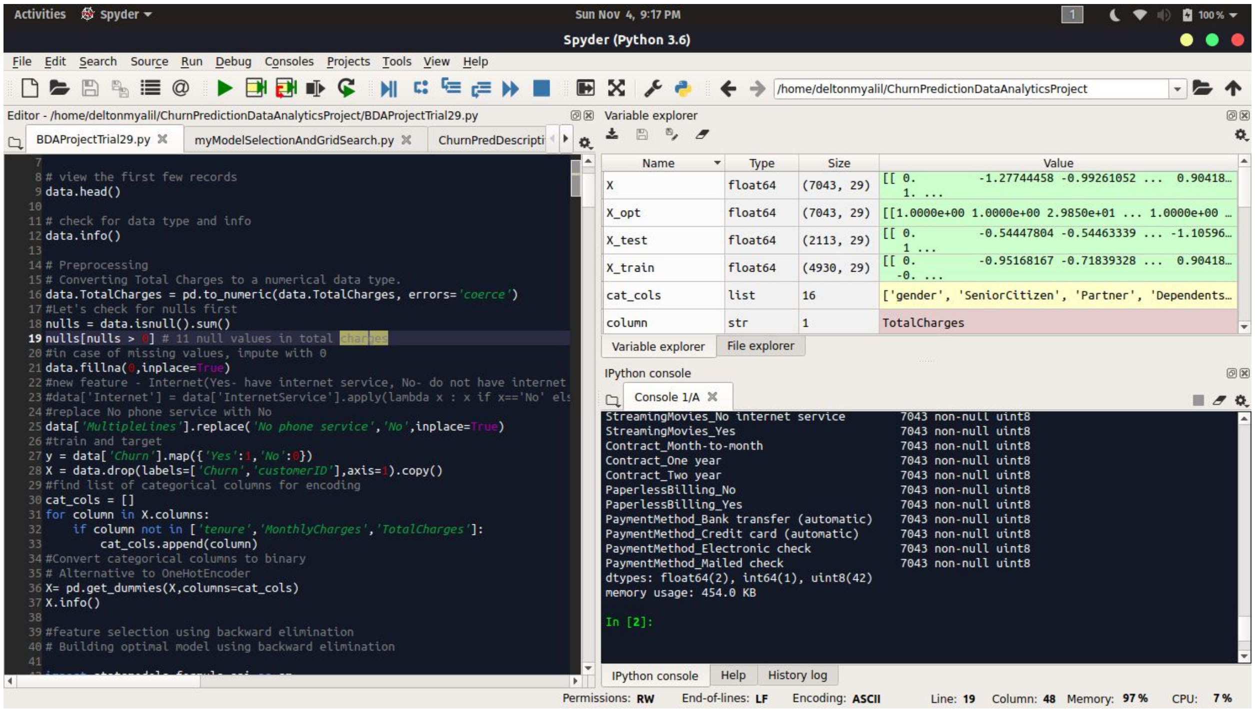
On a final note is that it is necessary to monitor the model every single day to check that it does not get outdated.

9. SCREENSHOTS, TABLES AND FIGURES

9.1 Tools and Software Used

9.1.1 Spyder:

We decided to code this project in python3. One of the best editors out there for python which is suited for data science is spyder. The features of spyder include separate in window panes for editor, console and file explorer. But what sets spyder apart from other Integrated Development Environments like Eclipse and Pycharm is that Spyder also comes with a handy Variable Explorer which can be used to see the intermediate variables which you get. Spyder also supports selective code running. You can select the part of the code you want to run. Spyder comes as a default software in Anaconda for WIndows, Mac and Linux.


Activities
Spyder
Sun Nov 4, 9:17 PM
100 %
Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
/home/deltonmyalil/ChurnPredictionDataAnalyticsProject
Editor - /home/deltonmyalil/ChurnPredictionDataAnalyticsProject/BDAProjectTrial29.py
BDAProjectTrial29.py
myModelSelectionAndGridSearch.py
ChurnPredDescripti
Variable explorer
Variable explorer
File explorer
IPython console
Console 1/A
IPython console
Help
History log

```

# view the first few records
data.head()

# check for data type and info
data.info()

# Preprocessing
# Converting Total Charges to a numerical data type.
data.TotalCharges = pd.to_numeric(data.TotalCharges, errors='coerce')
#Let's check for nulls first
nulls = data.isnull().sum()
nulls[nulls > 0] # 11 null values in total charges
#in case of missing values, impute with 0
data.fillna(0,inplace=True)
#new feature - Internet(Yes- have internet service, No- do not have internet
#data['Internet'] = data['InternetService'].apply(lambda x : x if x=='No' els
#replace No phone service with No
data['MultipleLines'].replace('No phone service','No',inplace=True)
#train and target
y = data['Churn'].map({'Yes':1,'No':0})
X = data.drop(labels=['Churn','customerID'],axis=1).copy()
#find list of categorical columns for encoding
cat_cols = []
for column in X.columns:
    if column not in ['tenure','MonthlyCharges','TotalCharges']:
        cat_cols.append(column)
#Convert categorical columns to binary
# Alternative to OneHotEncoder
X= pd.get_dummies(X,columns=cat_cols)
X.info()

#feature selection using backward elimination
# Building optimal model using backward elimination

```

Name	Type	Size	Value
X	float64	(7043, 29)	[[0. -1.27744458 -0.99261052 ... 0.90418… 1. ...
X_opt	float64	(7043, 29)	[[1.0000e+00 1.0000e+00 2.9850e+01 ... 1.0000e+00 …
X_test	float64	(2113, 29)	[[0. -0.54447804 -0.54463339 ... -1.10596… 1 ...
X_train	float64	(4930, 29)	[[0. -0.95168167 -0.71839328 ... 0.90418… -0. ...
cat_cols	list	16	['gender', 'SeniorCitizen', 'Partner', 'Dependents…
column	str	1	TotalCharges

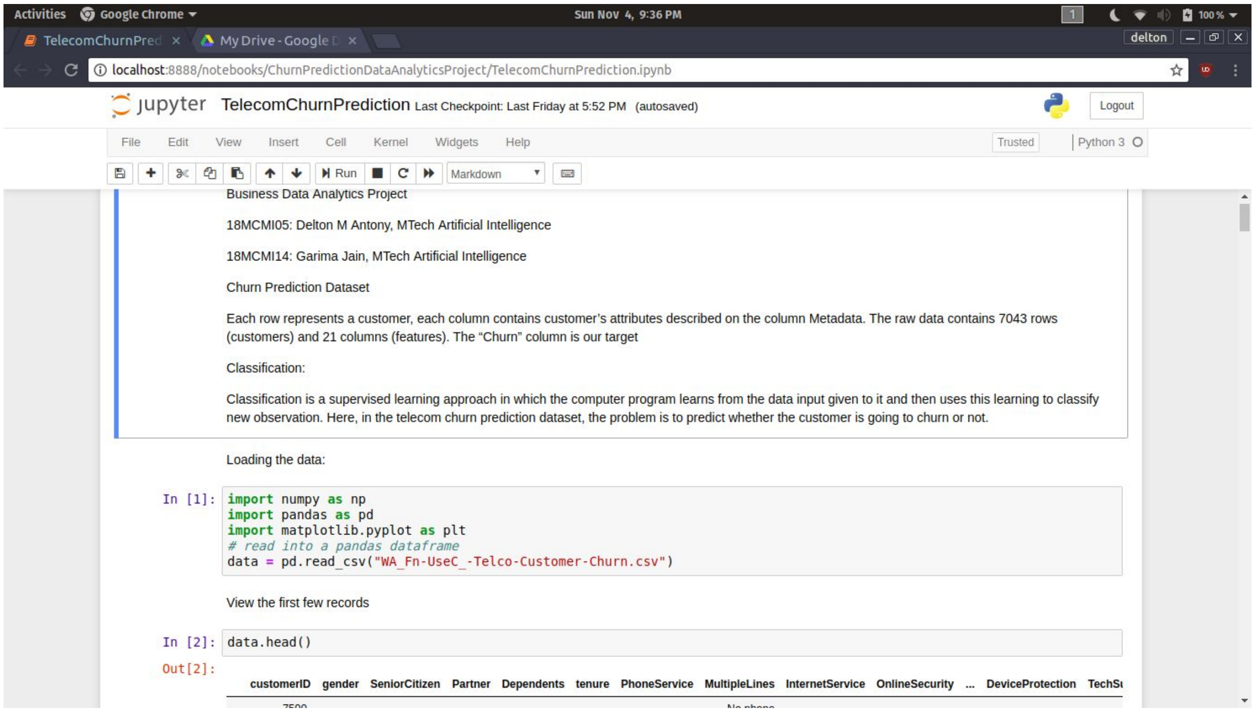
```
StreamingMovies_No internet service        7043 non-null uint8
StreamingMovies_Yes                        7043 non-null uint8
Contract_Month-to-month                    7043 non-null uint8
Contract_One year                          7043 non-null uint8
Contract_Two year                          7043 non-null uint8
PaperlessBilling_No                        7043 non-null uint8
PaperlessBilling_Yes                       7043 non-null uint8
PaymentMethod_Bank transfer (automatic)    7043 non-null uint8
PaymentMethod_Credit card (automatic)      7043 non-null uint8
PaymentMethod_Electronic check             7043 non-null uint8
PaymentMethod_Mailed check                 7043 non-null uint8
dtypes: float64(2), int64(1), uint8(42)
memory usage: 454.0 KB

In [2]:
```

Permissions: **RW** End-of-lines: **LF** Encoding: **ASCII** Line: **19** Column: **48** Memory: **97 %** CPU: **7 %**

9.1.2 Jupyter Notebook:

In order to make a notebook file which contains both computer readable code, graphs and tables and text which describes the code and the graphs, we used Jupyter Notebook. Once the coding was complete in Spyder, we took the code block by block and executed them in Jupyter Notebook. After each logical block of code, we described the code and the output generated by it - whether it be returns or tables and graphs. You can easily save the code, its results and descriptions as an executable ipynb file or a static html, tex or pdf file. Jupyter Notebook also comes preinstalled with Anaconda. Jupyter Notebook is available for Windows, Mac and Linux.



Business Data Analytics Project

18MCMI05: Delton M Antony, MTech Artificial Intelligence

18MCMI14: Garima Jain, MTech Artificial Intelligence

Churn Prediction Dataset

Each row represents a customer, each column contains customer's attributes described on the column Metadata. The raw data contains 7043 rows (customers) and 21 columns (features). The "Churn" column is our target

Classification:

Classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. Here, in the telecom churn prediction dataset, the problem is to predict whether the customer is going to churn or not.

Loading the data:

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        # read into a pandas dataframe
        data = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

View the first few records

```
In [2]: data.head()
Out[2]:
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSu

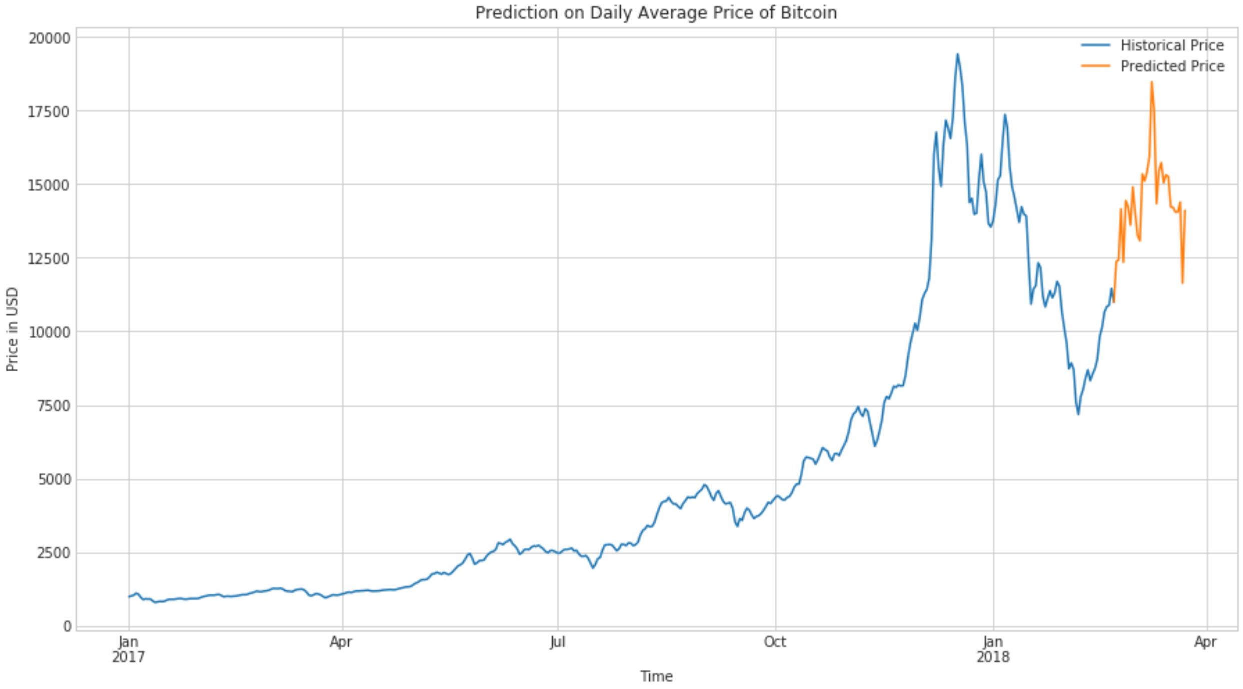
9.1.3 Development Environment:

I developed this project in my laptop running the latest Linux distribution of Ubuntu. The hardware used is i5 processor with 4GB RAM.

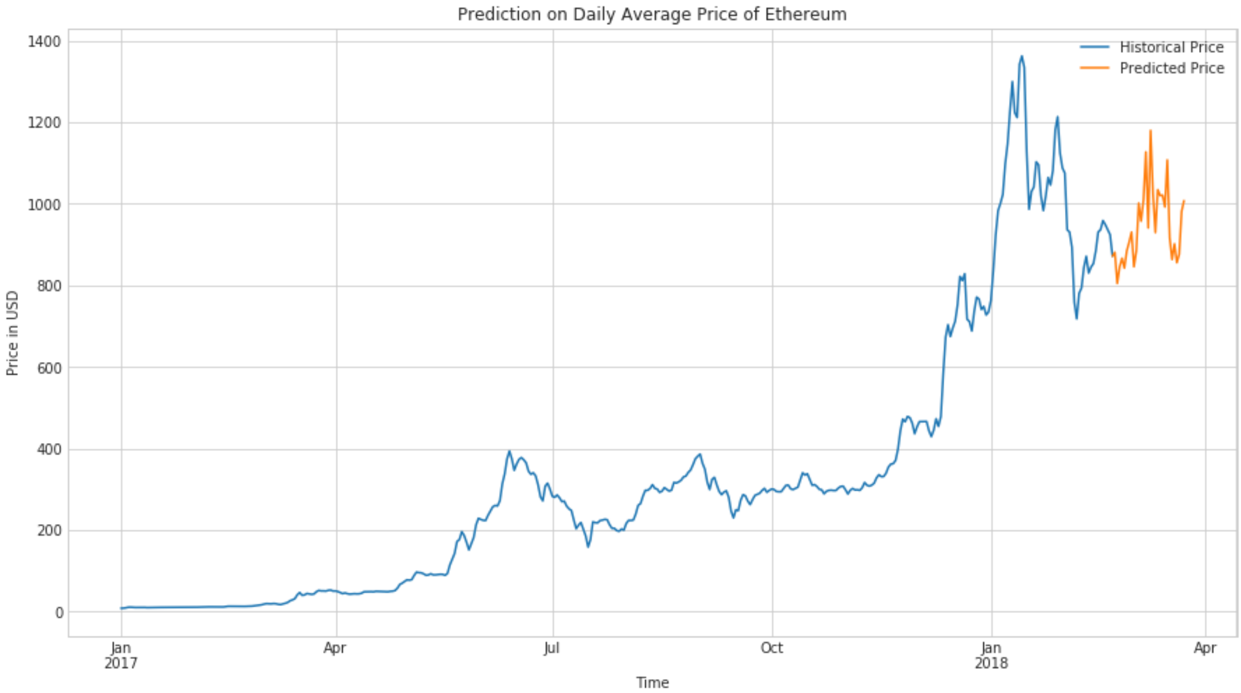
9.2 Result Figures

9.2.1 Prediction of the three cryptocurrencies:

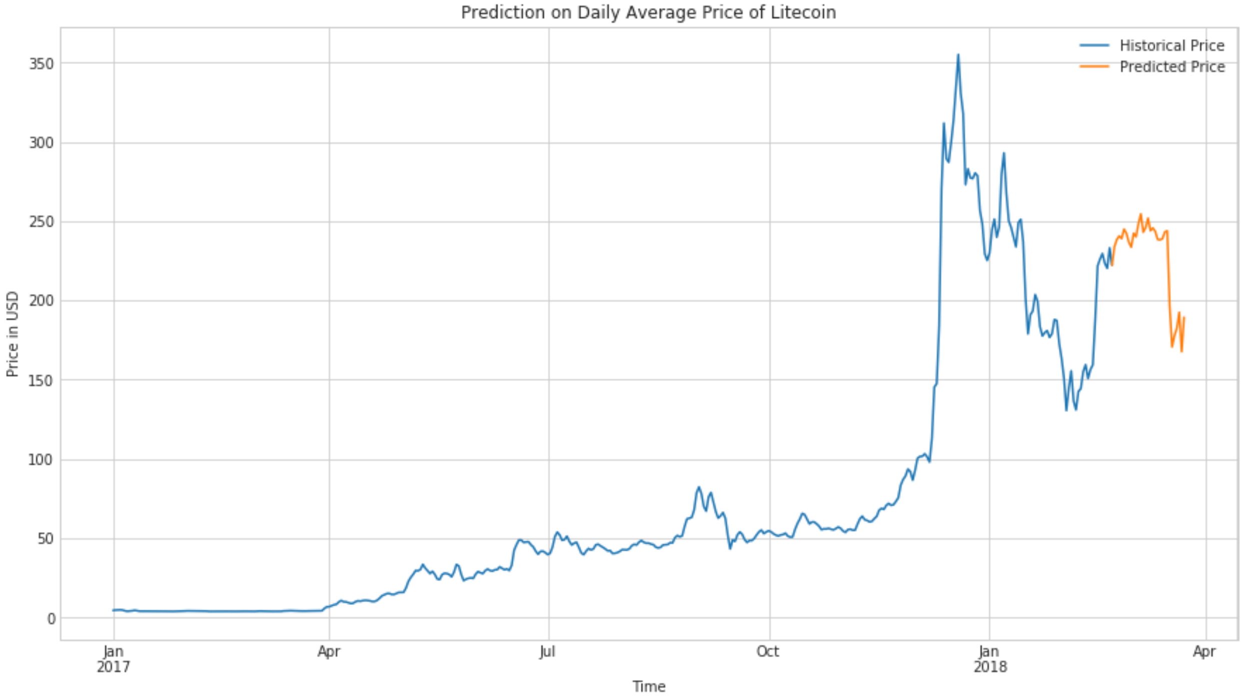
9.2.1.1 For Bitcoin:



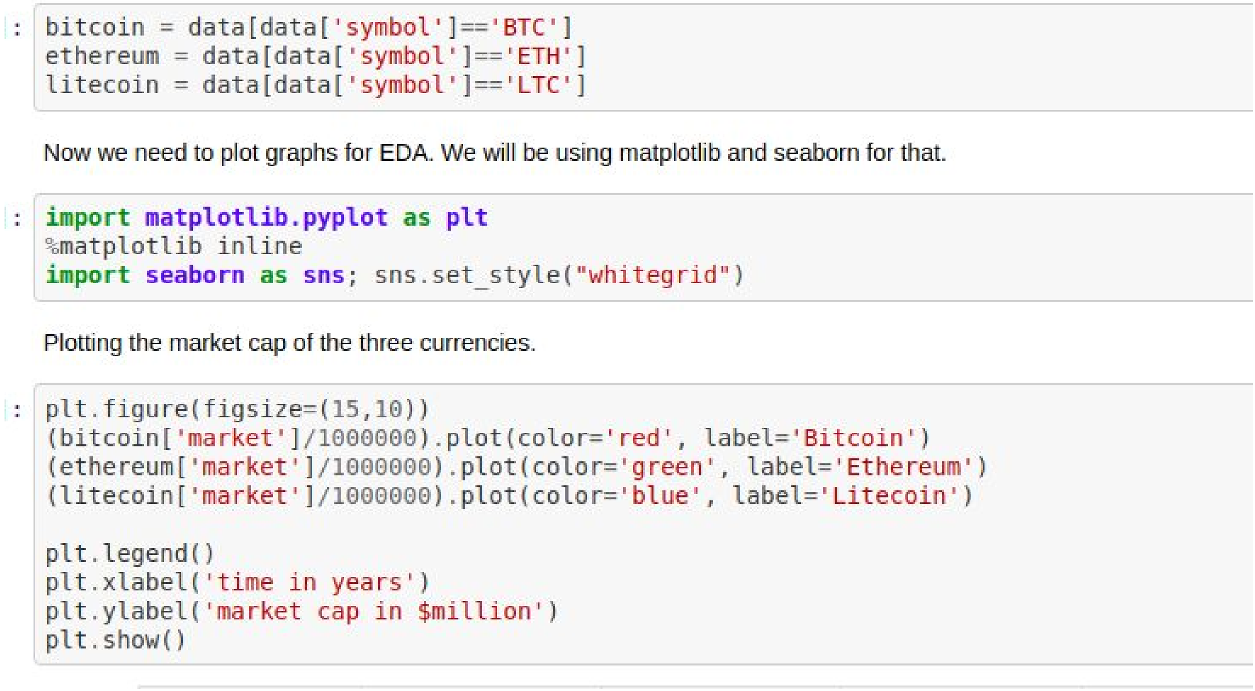
9.2.1.2 For Ethereum:



9.2.1.3 For Litecoin:



8.2.3 Screenshots of Python3 Code:



```
bitcoin = data[data['symbol']=='BTC']
ethereum = data[data['symbol']=='ETH']
litecoin = data[data['symbol']=='LTC']
```

Now we need to plot graphs for EDA. We will be using matplotlib and seaborn for that.

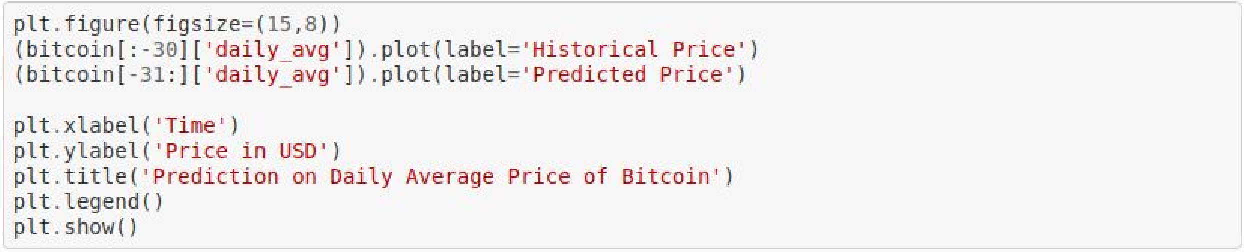
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns; sns.set_style("whitegrid")
```

Plotting the market cap of the three currencies.

```
plt.figure(figsize=(15,10))
(bitcoin['market']/1000000).plot(color='red', label='Bitcoin')
(ethereum['market']/1000000).plot(color='green', label='Ethereum')
(litecoin['market']/1000000).plot(color='blue', label='Litecoin')

plt.legend()
plt.xlabel('time in years')
plt.ylabel('market cap in $million')
plt.show()
```

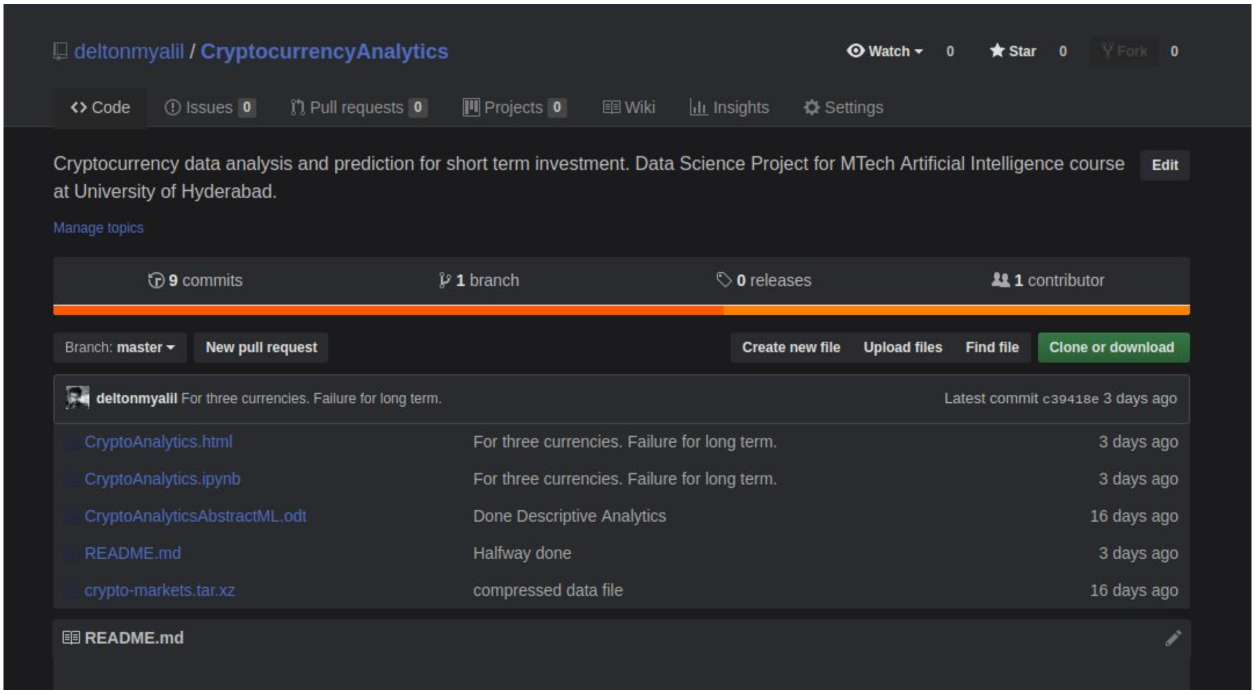
The above screenshot shows how matplotlib.pyplot and seaborn are used inline to display the graphs.



```
plt.figure(figsize=(15,8))
(bitcoin[:-30]['daily_avg']).plot(label='Historical Price')
(bitcoin[-31:]['daily_avg']).plot(label='Predicted Price')

plt.xlabel('Time')
plt.ylabel('Price in USD')
plt.title('Prediction on Daily Average Price of Bitcoin')
plt.legend()
plt.show()
```

The above screencap shows how I plotted the prediction and the actual price with respect to time.



Cryptocurrency data analysis and prediction for short term investment. Data Science Project for MTech Artificial Intelligence course at University of Hyderabad.

deltonmyalil For three currencies. Failure for long term. Latest commit c39418e 3 days ago

CryptoAnalytics.html	For three currencies. Failure for long term.	3 days ago
CryptoAnalytics.ipynb	For three currencies. Failure for long term.	3 days ago
CryptoAnalyticsAbstractML.odt	Done Descriptive Analytics	16 days ago
README.md	Halfway done	3 days ago
crypto-markets.tar.xz	compressed data file	16 days ago

This shows my code repository where you can find this project.

The link to the repository in Github is:
https://github.com/deltonmyalil/CryptocurrencyAnalytics

The git clone link is:
https://github.com/deltonmyalil/CryptocurrencyAnalytics.git

10. REFERENCES

[1]https://hackernoon.com/dont-be-fooled-deceptive-cryptocurrency-price-predictions-using-deep-learning-bf27e4837151

[2]https://dashee87.github.io/deep%20learning/python/predicting-cryptocurrency-prices-with-deep-learning/

[3]https://medium.com/activewizards-machine-learning-company/bitcoin-price-forecasting-with-deep-learning-algorithms-eb578a2387a3

[4]https://www.kaggle.com/jessevent/all-crypto-currencies