Executive Summary:

Cryptocurrency is the new investment rave at the moment and currently we are curious about the how it works, how it is used and what is the investment opportunities involved in investing in this type of commodity.

The purpose of this project is to create a basic economic prediction model to estimate the future trend of bitcoin. The presentation will include functions to input bitcoin purchases at different prices from hour to hour, day to day, month to month and year to year. The data is loaded in xlsx text and csv file format with different parameters such as purchase dates, purchase prices, open prices, high prices, low prices, purchase volume, purchase volume from, purchase volume to, time series and percent changes of price during a period of time.

We plan to analyze the bitcoin data looking at low and highs prices per day, month, and year using NumPy arrays, Matplotlib, and Scipy. We will use these libraries to analyzed the data to train our models with analytical tools such as Linear Regression, Time Series, Machine Learning (Long Short Term Memory (LSTM)) and Tensorflow. We will test our models using these tools to make prediction for the future trends of bitcoin.

Analyzing and Data Wrangling:

As we clean the data from our hour by hour, day to day, month to month, and year to year we removed commas, percent signs and manipulated the date to make the xslx and csv file more readable in Jupyter. When uploading the files into Jupyter we checked each data set and looked up the max and min values of the parameters. This data allowed us to find the date of these events and make meaningful graphs to visually plot the moving prices of the different cryptocurrencies to show the volatility and stabilities of the coins in a time series graph.

We tested our data set in different plots such as line graph, box plot, pairplot, histograms, sma, cma, and heatmaps, ARIMA models.

Environment used to Setup and run the Project:

The project is developed using the following environment:

Mac

Python 3.6.2 Andaconda custom (64-bit)

Referencing packages

numpy	
matplotlib	2.1.0
scipy	0.19.1
pandas	0.20.3
seaborn	0.8.0
scikit learn	0.19.1
keras	2.2.0
plotly	2.6.0
cufflinks	0.13.0

Running the Project:

Step 1
Load all the dataset in to jupyter.
List of the dataset:

BitcoinHistoricalDataByDay.xlsx
BitcoinHistoricalDataByDay2010to2018.xlsx
BitcoinHistoricalDataByWeek.xlsx
BitcoinHistoricalDatabyMonth.xlsx
Coinbase_BTCUSD_1h.xlsx
Coinbase_ETHUSD_1h.csv
Coinbase_LTCUSD_1h.csv

Step 2
After loading the dataset
Run the jupyter note book titled
“Bitcoin_Python_part.ipynb”

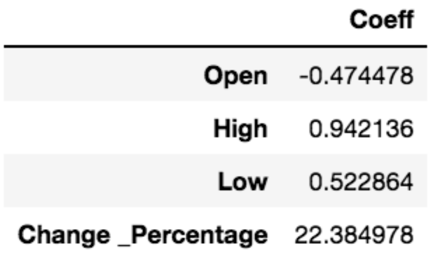
Step 3 Run on Machine Learning part of the project
Run the jupyter notebook titled

“Bitcoin_ML_ISLM_part.ipynb”

Model Performance and Evaluation:

For basic analysis we used seaborn and plotly’s iplot feature to analyze the price trend. Then we proceeded with some statistical analysis using statsmodel. Timeseries and ARIMA to observe the trend over a year with a weekly window on the axis. We used two Machine Learning models – Linear Regression and RNN with LSTM(Long Short Term Memory) method for prediction.

Our linear regression model was using BitcoinHistoricalDataByDay2010to2018.xlsx dataset that contains daily rate from early 2010 to 6/14/2018. We split the training and testing data to 70/30 ratio, and using the linear regression lmfeature from PythonScikit-learn module (sklearn.linear_model) and overall has an accuracy rate of 99% success rate.



	Coeff
Open	-0.474478
High	0.942136
Low	0.522864
Change _Percentage	22.384978

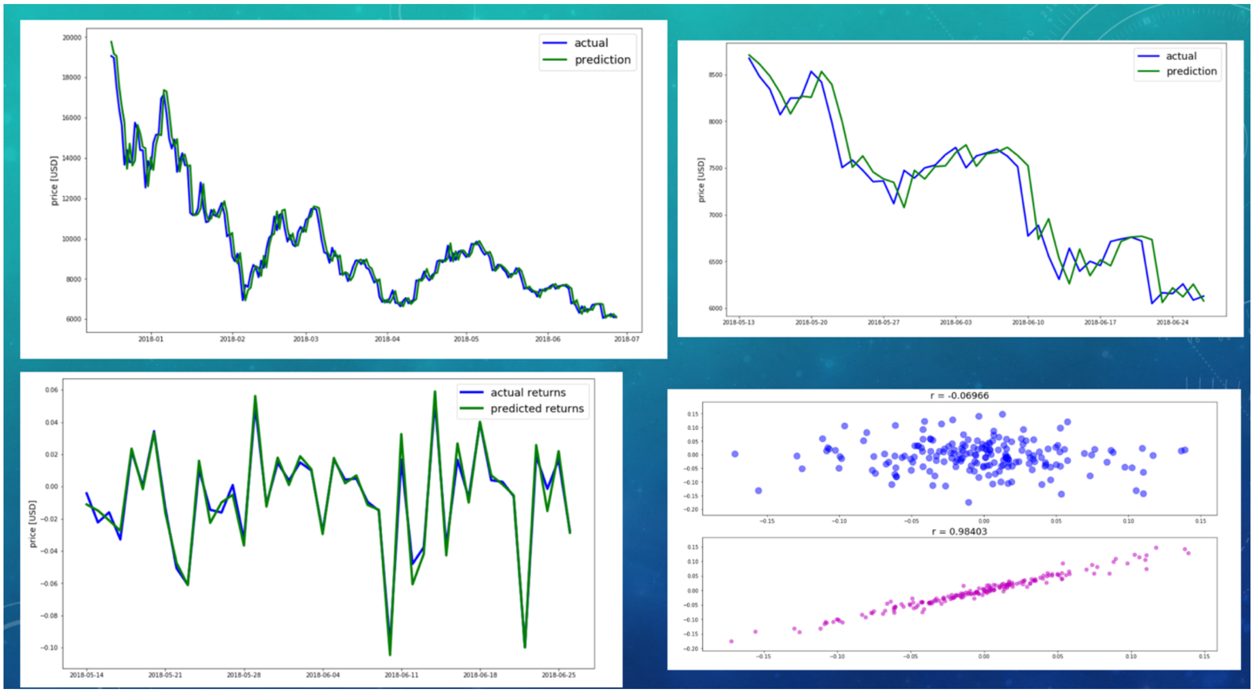
Our dataset used for Recurrent Neuro Network using LSTM model is using the API of Cryptocompare.com's historical data up to the current day. We used Keras to carry out implementation phase in which the training/test is 80/20 split, used Adam optimizer as the Algorithm for LSTM because it's straightforward and computationally efficient, requiring Little memory and very well suited for non- stationary objectives like cryptocurrency price movement. We set the parameters as follows:

```
# data parameters
window_len = 7
test_size = 0.1
zero_base = True

# model parameters
lstm_neurons = 20
epochs = 50
batch_size = 4
loss = 'mae'
dropout = 0.25
optimizer = 'adam'
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 20)                2160
_________________________________________________________________
dropout_1 (Dropout)          (None, 20)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 21
_________________________________________________________________
activation_1 (Activation)    (None, 1)                 0
=================================================================
Total params: 2,181
Trainable params: 2,181
Non-trainable params: 0
```

Overall we have **98.8**% model performance accuracy rate for the bitcoin price trend prediction, and we predicted the bitcoin price will go below $6,000 in the next month.


actual
prediction
price [USD]
20000
18000
16000
14000
12000
10000
8000
6000
2018-01
2018-02
2018-03
2018-04
2018-05
2018-06
2018-07
actual
prediction
price [USD]
8500
8000
7500
7000
6500
6000
2018-05-13
2018-05-20
2018-05-27
2018-06-03
2018-06-10
2018-06-17
2018-06-24
actual returns
predicted returns
price [USD]
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
2018-05-14
2018-05-21
2018-05-28
2018-06-04
2018-06-11
2018-06-18
2018-06-25
r = -0.06966
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
r = 0.98403
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15