Time Series

Cryptocurrency, over the years, have gained interest from investors, central banks and governments all over the world. The lack of political regulation and their market far from being efficient, require new forms of regulation in the future. From an econometric viewpoint, the process underlying the evolution of the cryptocurrencies volatility has been found to exhibit at the same time differences and similarities with other financial time series, e.g. foreign exchanges returns, stock market, commodity market. We investigate the effect of accounting for long memory in the volatility process as well as its asymmetric reaction to past values of the series to predict volatility levels.

To measure the volatility levels in data, time series analysis is a perfect tool. Time series analysis comprise methods for analyzing time series data in order to extract meaningful characteristics and volatility levels of the data. Perfect mechanism to predict future values based on previously observed data, monitoring trends and even feedback and feedforward control, all could be achieved from time series analysis. Many of the principal attributes that characterize generally financial time–series also apply to cryptocurrencies. For instance cryptocurrencies exhibits, similar to other financial markets; time-varying volatility, extreme observations and an asymmetric reaction of the volatility process to the sign of past observations (i.e., leverage effect).

Autocorrelation

In the case of time series data, if the observations show inter-correlation, specifically in those cases where the time intervals are small, then these inter-correlations are given the term of autocorrelation. Autocorrelation is a mathematical representation for finding repeating patterns, such as the presence of a periodic signal (repeated regular interval) complicated by noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. It is the same as calculating the correlation between two different time series, except that the same time series is used twice: once in its original form and once lagged one or more time periods.

Autocorrelation can be useful for technical analysis, which is more concerned with the trends of and relationships between crypto prices and value. Technical analysts can use autocorrelation to see how much of an impact past value for a security have on its future price.

EXPLAIN BITCOIN RESULT

|  |  |  |
| --- | --- | --- |
|  | Bitcoin | Ethereum |
| Random Forest Regressor | R2: 0.85  MAE: 1010.13  MSE: 2540545 | R2: 0.84  MAE:66.67  MSE:8603.4 |
| Gradient Boosting | R2: 0.86  MAE: 1016.9  MSE: 2464682.2 | R2: 0.82  MAE: 70.71  MSE: 9799.4 |
| Extra Trees | R2: 0.87  MAE: 900.37  MSE: 2222611.46 | R2: 0.81  MAE: 70.1  MSE: 10288.24 |
| Bayesian Ridge | R2: 0.7  MAE: 1714.85  MSE: 5178364.12 | R2: 0.47  MAE: 106.9  MSE: 29131.3 |
| Elastic Net CV | R2: 0.66  MAE: 1831.29  MSE: 5841023.25 | R2: 0.35  MAE: 143.7  MSE: 36031.04 |

Applying machine learning models

In this coursework we have managed to use five different kinds of regression models that were applied to Bitcoin and Ethereum. As we can see from the table above the five models used in this coursework were: Random Forest Regressor, Gradeient Boostin, Extra Trees, Bayesian Ridge and Elastic Net CV.

The R-squared value of a regression function shows how close the data is to the fitted regression line This means that in Bitcoin, the regression function Extra Trees had 87% of the data closely plotted against the fitted line. The Mean Absolute Error (MAE) can be defined as the measure of difference between two continuous variables. Simply put Mean Absolute Error (MAE) is the average vertical distance between each point and the Y=X line. This value has to as low as possible for the regression to be claimed as suitable for a model. The MSE (Mean Square Error) measures the average of square of errors. In this case the MSE is 2222611.46. In Bitcoin, we chose Extra Trees as our suitable regression function.

In Ethereum however, we have decided to go with Random Forest Regression. The R-squared value tells us that 84% of the points are close to the fitted regression line. The MAE value is 66.67 while the MSE is 8603.4 which tells us the average of square of errors. It can simply be defined as the difference between estimator and the estimated.

Price prediction

|  |
| --- |
| def prediction(name, X, y, X\_forecast):  if name in ['BTC', 'ETH']:  model = RandomForestRegressor(n\_estimators=200)  else:  model = ExtraTreesRegressor(n\_estimators=500, min\_samples\_split=5)  model.fit(X, y)  target = model.predict(X\_forecast)  return target |

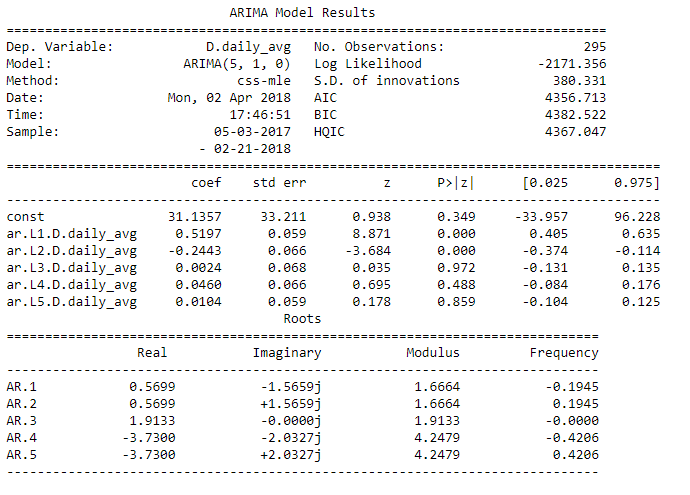
The prediction model was not accurate enough as they have not regarded the time series nature of the data.

In order to improve the prediction we need to implement ARIMA. ARIMA stands for Autoregressive Integrated moving averages. ARIMA models is used in time series data to have a better understanding of the data and to further predict the future points of the series.

This information will help investors to predict the future exchange rate of Bitcoin and in the same time volatility need to be monitor closely. This action will help investors to gain better profit and reduce loss in investment decision.

Volatility is a statistical measure of the dispersion of returns for a stock market. Volatility of stock markets has created much attention among investors because high volatility can bring high returns or losses to investors (Abu Bakar and Rosbi, 2017). This situation creates a risk to investors, because a rational investor always makes an investment decision based on risk and return (Lee, et al., 2016).

|  |
| --- |
| ## Explain and Predict ARIMA  from statsmodels.tsa.arima\_model import ARIMA  #model = ARIMA(bitcoin[0:-30], order=(43,1,0))  model = ARIMA(bitcoin[0:-30], order=(5,1,0))  model\_fit = model.fit()  print(model\_fit.summary())  # plot residual errors  residuals = pd.DataFrame(model\_fit.resid)  residuals.plot()  plt.show()  residuals.plot(kind='kde')  plt.show()  print(residuals.describe()) |



The above results show that total number of observations are 295. The sample selected is from 3rd May 2017 till 21st February 2018. The log likelihood is -2171.3. the z value is 0.938 of constant which means 0.9 standard deviations away from mean. While the p value is 0.349 which is the probability that we have falsely rejected the null hypothesis.

Stationary time series has constant mean, variance and other statistical properties over time. In-order to check for stationarity in time series models, augmented Dicky-Fuller tests are run. This determines how strongly a time series is defined by a trend. It uses autoregressive model and optimizes an information criterion across multiple different lag values.

An augmented Dickey–Fuller test (ADF) tests the [null hypothesis](https://en.wikipedia.org/wiki/Null_hypothesis) that a [unit root](https://en.wikipedia.org/wiki/Unit_root) is present in a [time series](https://en.wikipedia.org/wiki/Time_series) [sample](https://en.wikipedia.org/wiki/Sample_(statistics)). The [alternative hypothesis](https://en.wikipedia.org/wiki/Alternative_hypothesis) is usually [stationarity](https://en.wikipedia.org/wiki/Stationarity_(statistics)) or [trend-stationarity](https://en.wikipedia.org/wiki/Trend_stationary).

1. from statsmodels.tsa.stattools
2. import adfuller X = bitcoin[0: -30]['daily\_avg'].values result = adfuller(X)
3. print('ADF Statistic: %f' % result[0])
4. print('p-value: %f' % result[1])
5. print('Critical Values:')
6. for key, value in result[4].items():
7. print('\t%s: %.3f' % (key, value))

The test shows that Bitcoin data present is stationary.