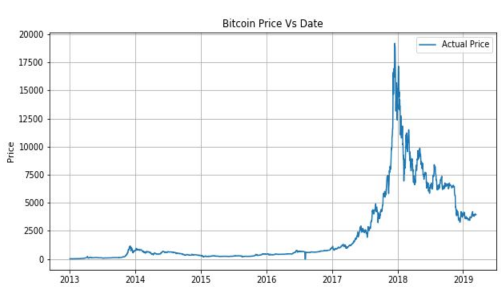
Predicting bitcoin price using monte carlo simulation and arima model.

ABSTRACT : Bitcoin is the one of the cryptocurrencies which offers several benefits like fast transaction speeds elimination of third-party intermediary to process. It is decentralized, open, widely accessible digital currency. In this paper we will use Monte-Carlo simulations and arima methods to predict with minimum error rates but it is hard to provide the precise predictions because of huge price oscillations.

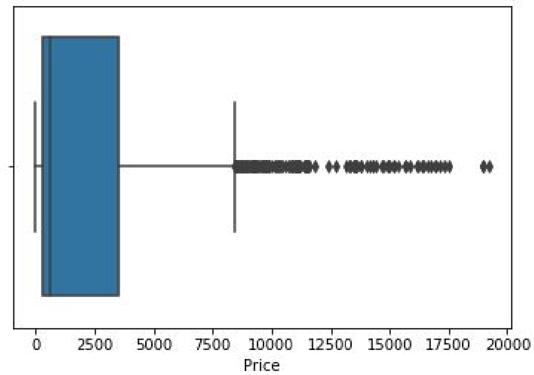
INTRODUCTION: Bitcoin is a peer to peer electronic cash which was introduced by the group of people named Satoshi Nakamoto in the year 2009. It is a form of digital currency and it is decentralized without any central banks, it is highly protected by cryptographic techniques. Bitcoin transactions mainly rely on blockchain technology where all the successful transactions are included. The integrity and chronological order of the blockchain are enforced with cryptography. Bitcoins are transferred from one person to another person through a process called mining. Bitcoin transactions can be done through bitcoin wallets that hold the seed which is useful for making the transactions securely from one to many in a single transaction and it takes hardly 10-20 minutes for a transaction which is very less time when compared to other transactions. The units of a bitcoin are milli-bitcoin (mBTC) and Satoshi (sat). It is the most popular among all other cryptocurrencies.

Bitcoin price data for the years 2013 to 2019 (14-Mar-2019) from www.investing.com website. Plotted the graph against date vs bitcoin price and observed that initially the price was very low in 2013 and from 2013 onwards the price of bitcoin was started increasing which can be seen in the fig

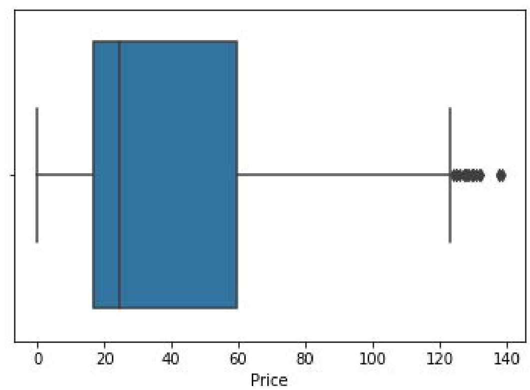


PREPROCESSING OF TIME SERIES DATA:

On observing the fluctuations and outliers in the above plotted graph so,there is a need to preprocess and do the transformations to dataset to reduce the effect of the outliers on the original values.This can be done by using Square Root Transformation [1] technique.



BEFORE SQUARE ROOT TRANSFORM



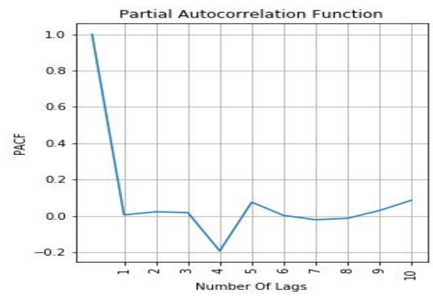
AFTER SQUARE ROOT TRANSFORM

Square root transformation handles the risk of outliers.Now the transformed time series data is checked for stationary i.e., constant variance using ADfuller hypothesis test(Augmented Dickey–Fuller test)[2].The transformed data fails in the ADfuller test there is a need to perform Following algorithm :

1. Lag the transformed data by one.
2. Find the difference between transformed data and lagged data.
3. Now check the above data for stationarity using the above mentioned hypothesis test.
4. Do the step 1-3 until it passes the test.

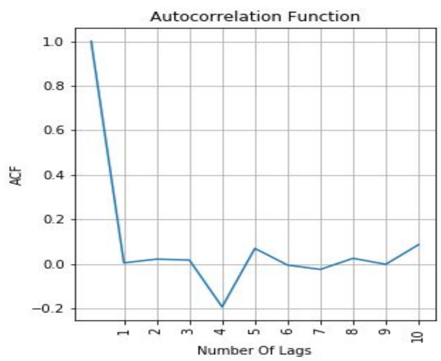
‘Now, the stationarized data is used to find the orders of AR, I, MA models i.e., O(AR),O(I),O(MA) respectively.

O(AR): Can be found using partial Autocorrelation functions (PACF)[3] which are imported from Package named statsmodels of python.



O(I): This is equal to the number of iterations performed in the above algorithm.

O(MA): Can be found using Autocorrelation functions (ACF)[4].



ARIMA MODEL:

<u>AR:</u>

Describes how present value is a function of previous value.

$$\mathbf{Y_t = c + \varphi_1 Y_{t-1} + e_t}$$

Where Y_t is a present and Y_{t-1} is a previous value at time t and t-1 respectively. e_t is a random error,c,φ_1 are constants.

<u>MA:</u>

Describes how present error is a function of previous error terms

$$\mathbf{Y_t = c + \theta_1 e_{t-1} + e_t}$$

Now arima model is imported from package named statsmodels of python.The Square Root transformed data is taken and then splitted into Training Data and Testing Data(last 7 days data).The arima model with order (1,1,1) is trained using training data,now we are predicting the bitcoin prices for next 7 days and calculating the error metrics like RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error).

MONTE CARLO SIMULATION

Monte Carlo method is based on the brownian motion of particles[5] which involves two steps a constant or drift and a random motion For the above raw data scan the dataset and update the values with the neighbouring values if their value is zero because there is a chance of getting divide by zero error .Make the dataset into training data and testing data .Like the brownian motion Monte carlo method requires both the drift and some random values for forecasting the prices of the bitcoin.

Tomorrows_stock_price=todays_stock_price * exp(r)

Where r is the periodic daily returns (PDR)

r = drift + random_stochastic_offset

Drift is like the constant in the brownian motion of the particles which tells Where to increase or where to decrease or the expected rate of change each in day that has the greatest odds of occurring and the random part combines with the drift to produce returns that are normally distributed

drift=mean-variance * 0.5

The calculation of drift includes the calculation of mean and variance of the logarithmic returns.

Logarithmic returns is the logarithmic ratio of the todays_stock_value and the yeasterdays_stock_value.

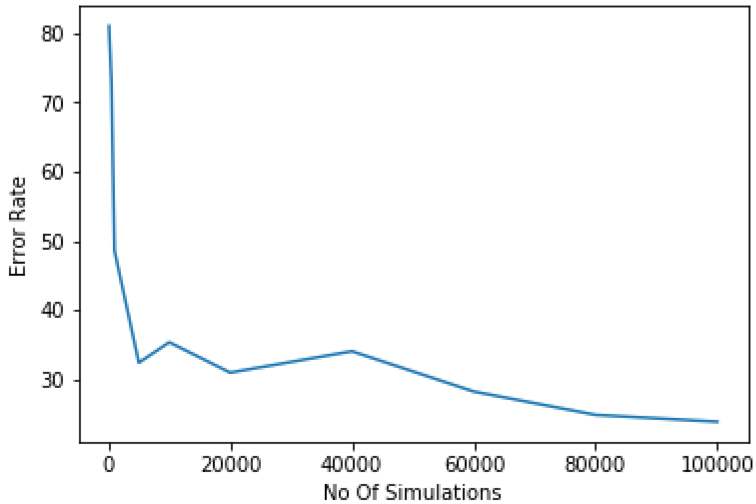
Random value can be any value in the range of [0,1) chosen by the random function.

Random_stochastic_offset will be the inverse normal distribution of the random value produced.

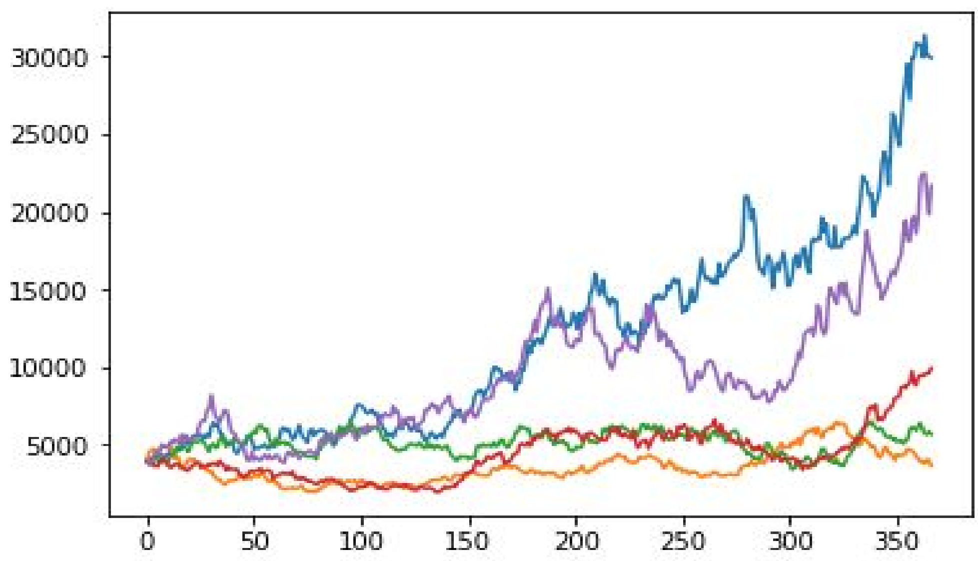
The Random_stochastic_offset will change in each simulation because of the random value produced.

Number of simulations = [100,500,1000,5000,10000,20000,40000,60000,80000,100000]

Error Rates = [80.99,72.85,48.58,32.40,35.35,30.97,34.05,28.24,24.87,23.88]



From the above results it is clear that as the number of simulations increases error rate decreases and at particular simulation value the change in error rate will be negligible and there we can stop the simulations.



The above graph is the predicted values of bitcoin prices for the next 365 days from now(14-MAR-2019).

Results:

As mentioned earlier Dataset is divided Into training set and test set.Now the and test data

GRAPH:

BAR GRAPH: