**ABSTRACT**

Crypto-currency such as Doge coin is more popular these days among investors. In the proposed work, it is studied to forecast the Doge coin price precisely considering different parameters that influence the Doge coin price. This study first handles, it is identified the price trend on day by day changes in the Doge coin price while it gives knowledge about Doge coin price trends. The dataset till current date is taken with open, high, low and close price details of Doge coin value. Exploiting the dataset machine learning module is introduced for prediction of price values. The aim of this work is to derive the accuracy of Doge coin prediction using different machine learning algorithm are LSTM, RMSE and Monte Carlo Simulation. A rely more on long-term history while LSTM tends to rely more on short-term dynamics, which indicate the efficiency of LSTM to utilize useful information hidden in historical memory is stronger prediction of proposed method of the system. Experiment results are showing the better performance of the system.

**Keywords:** Doge coin, Cryptocurrency, Machine Learning, Artificial Intelligence, Data Science, Prediction, LSTM, Monte Carlo Simulation.

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

A crypto currency is a digital form of currency that was originally made as regular means of exchange. Crypto currency is the peer-to-peer digital moneyory and payment system that exist online via a controlled algorithm. When a miner cracks an algorithm to record a block of transactions to public ledger named block chain and the crypto currency is created when the block is added to the block chain. It allows people to store and transfer through encryption protocol and distributed network [1]. Mining is a necessary and competitive component of the crypto currency system. The miner with more computational power has a better chance of finding a new coin than that of less [2]. Bit coin is the first and one of the leading digital currencies (its market capitalization had more than $ 7 billion in 2014, and then it increased significantly to $ 29 billion in 2017) [3, 4], which was first introduced by Satoshi Nakamoto in 2008. Among many features of bit coin, the most impressive one is decentralization that it can remove the involvement of traditional financial sectors and monetary authorities effectively due to its block chain network features [4]. In addition, the electronic payment system of Bit coin is based on cryptographic proof rather than the trust between each other as its transaction history cannot be changed unless redoing all proof of work of all block chain, which play a critical role of being a trust intermediary and this can be widely used in reality such as recording charitable contribution to avoid corruption. Moreover, bit coin has introduced the controllable anonymity scheme, and this enhances users’ safety and anonymity by using this technology, for instance, we can take advantage of this property of block chain to make identification cards, and it not only can protect our privacy but verify our identity.

Crypto currencies have been introduced to the market in 2008 when Satoshi Nakamoto rendered the first crypto currency named Bit coin, through his white paper Bit coin: peer to- peer Electronic Cash System intending to replace the old and widely used economic system. The traditional cash exchange system is entirely dependent on the centralized third-party financial institutions such as banks in order to validate and execute the monetary transactions. It is restricted to perform only amount of money and possesses various issues such as trust, security, transparency, and flexibility. To overcome the aforementioned issues, a peer-to-peer (P2P) network-based currency namely crypto currency is a perfect solution. It eliminates the involvement of centralized third-party banking systems. In this, each transaction has to be acknowledged (validated) by each peer using any suitable consensus algorithm such as proof of work (PoW), proof of stack (PoS), and many more. This algorithm internally uses the transaction hash and timestamps for transaction confirmation. It requires peers in the network to solve a complex mathematical problem to add a new block in the chain. Mining is the process of solving the problem, and miners get rewarded for their work through crypto currency. Bit coin is a crypto currency which is used worldwide for digital payment or simply for investment purposes. Bit coin is decentralized i.e. it is not owned by anyone. Transactions made by Bit coins are easy as they are not tied to any country. Investment can be done through various marketplaces known as “bit coin exchanges”. These allow people to sell/buy Bit coins using different currencies. The largest Bit coin exchange is Mt Gox. Bit coins are stored in a digital wallet which is basically like a virtual bank account. The record of all the transactions, the timestamp data is stored in a place called Block chain. Each record in a blockchain is called a block. Each block contains a pointer to a previous block of data. The data on blockchain is encrypted. During transactions the user’s name is not revealed, but only their wallet ID is made public. Nowadays, investing in crypto currencies, like Bit coin, is one of the efficient ways of earning money. For example, the rate of Bit coin significant rises in 2017, from a relatively low point 963 USD on January 1ST 2017, to its peak 19186 USD on December 17th 2017, and it closed with 9475 USD at the end of the year. Consequently, the rate of return of bit coin investment for 2017 was over 880% [5], which is an impressive and surprising scenery for most investors. While an increasing number of people are making investments in Crypto currency, the majority of investors cannot get such profit for being inconsiderable to crypto currencies’ dynamics and the critical factors that influence the trends of bit coins. Therefore, raising people’s awareness of vital factors can help us to be wise investors. Although market prediction is demanding for its complex nature [6, 7], the dynamics are predictable and understandable to some degree. For example, when there is a shortage of the bit coin, its price will be increased by their sellers as investors who regard bitcoin as a profitable investment opportunity will have a strong desire to pay for bitcoin. Furthermore, the price of bit coin may be easily influenced by some influential external factors such as political factors of the system.

* 1. **EXISTING SYSTEM**

The price of the bitcoin can be predicted using the twitter sentiment analysis which is done by analyzing the sentiment score with the historical price to predict the future price. The major contribution is a sentiment analyzer which can distinguish between the positive and negative tweets of the bitcoin over Twitter. Though it is possible to predict the price of the bitcoin from the twitter sentimental analysis, it is comparatively difficult to know the positive and negative views in other social platforms. The bitcoin price prediction using machine learning algorithms such as the Bayesian Regression and GLM/Random forest which provides less accuracy so the deep learning algorithm such as Long short term Memory algorithm is used for the predicting the price of the bitcoin which provides the better accuracy than the machine learning techniques but this is used only for the short term dependencies.

* 1. **PROPOSED SYSTEM**

The proposed work is implemented in Python 3.6.4 with libraries scikit-learn, pandas, matplotlib and other mandatory libraries. We will develop a financial data predictor program in which there will be a dataset storing all historical doge coin prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making. We propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. The main aim of our project to predict the level of doge coin in each every year to show the result based on the datas of open, close, high, low, adjust close and Total volume of our datas prediction of the system. In our proposed, we use some steps to show the doge coin or bitcoin price prediction each and every year. The steps are Dataset collection, preprocessing, data visualization model show each and every data result of the system. Finally, we use the LSTM and Monte Carlo simulation algorithm is used to show the prediction of our result as well as show the accuracy of our model. Time series forecasting models can predict future values based on previous, sequential data using LSTM and Monte Carlo simulation. This increases the accuracy of demand forecasters, resulting in better business decisions. The benefits of the proposed method include improved system performance, accuracy, stability, year-to-year prediction, efficiency, robustness, and flexibility.

**1.4 STATEMENT OF PROBLEM**

* In the previous state of the system, the problem statement described the prediction of doge coin or bit coin outcomes. In the previous stage, we can use approaches such as Support Vector Machine (SVM) and Random Forests (RF) in the system's bit coin or doge coin prediction.
* The disadvantages of our outcomes are low performance, incorrect prediction, and an unstable system.
  1. **LITERATURE REVIEW**

[1] G. Giudici et al., “Cryptocurrencies: Market analysis and perspectives”. The papers in this special issue focus on the emerging phenomenon of crypto currencies. Cryptocurrencies are digital financial assets, for which ownership and transfers of ownership are guaranteed by a cryptographic decentralized technology. The rise of crypto currencies’ value on the market and the growing popularity around the world open a number of challenges and concerns for business and industrial economics. Using the lenses of both neoclassical and behavioral theories, this introductory article discusses the main trends in the academic research related to crypto currencies and highlights the contributions of the selected works to the literature. A particular emphasis is on socio-economic, misconduct and sustainability issues. We posit that crypto currencies may perform some useful functions and add economic value, but there are reasons to favor the regulation of the market. While this would go against the original libertarian rationale behind crypto currencies, it appears a necessary step to improve social welfare.

[2] S. Khuntit et al., “Adaptive market hypothesis and evolving predictability of bitcoin”. This study evaluates the adaptive market hypothesis (AMH) and evolving return predictability in bitcoin market. We use two robust methods in a rolling-window framework to capture time-varying linear and nonlinear dependence in bitcoin returns. We find that market efficiency evolves with time and validates the AMH in bitcoin market. We evaluate the adaptive market hypothesis (AMH) in Bitcoin market. Linear and nonlinear dependence checked using rolling-window approach. Efficiency evolves with emergence of events and adheres to the AMH proposition.

[3] T. Phaladisailoed et ., “Machine learning models comparison for bitcoin price prediction”. In recent years, Bitcoin is the most valuable in the cryptocurrency market. However, prices of Bitcoin have highly fluctuated which make them very difficult to predict. Hence, this research aims to discover the most efficient and highest accuracy model to predict Bitcoin prices from various machine learning algorithms. By using 1-minute interval trading data on the Bitcoin exchange website named bitstamp from January 1, 2012 to January 8, 2018, some different regression models with scikit-Iearn and Keras libraries had experimented. The best results showed that the Mean Squared Error (MSE) was as low as 0.00002 and the R-Square (R 2 ) of the system.

[4] G. Sera et al., “Sentiment-driven price prediction of the bitcoin based on statistical and deep learning approaches”. Nowadays, Bitcoin has become the most popular crypto currency, which gains the attention of investors and speculators alike. Asset pricing is a risky and challenging activity that enchants lots of shareholders. Indeed, the difficulty in making predictions lies in understanding the multiple factors that affect the Bitcoin price trend. Modeling the market behavior and thus, the sentiment in the Bitcoin ecosystem provides an insight into the predictions of the Bitcoin price. While there are significant studies that investigate the token economics based on the Bitcoin network, limited research has been performed to analyze the network sentiment on the overall Bitcoin price. In this paper, we investigate the predictive power of network sentiments and explore statistical and deep-learning methods to predict Bitcoin future price. In particular, we analyze financial and sentiment features extracted from economic and crowd-sourced data respectively, and we show how the sentiment is the most significant factor in predicting Bitcoin market stocks. Next, we compare two models used for Bitcoin time-series predictions: the Auto-Regressive Integrated Moving Average with eXogenous input (ARIMAX) and the Recurrent Neural Network (RNN). We demonstrate that both models achieve optimal results on new predictions, with a mean squared error lower than 0.14%, due to the inclusion of the studied sentiment feature. Besides, since the ARIMAX achieves better predictions than the RNN, we also prove that, with just a linear model, we may obtain outstanding market forecasts in the Bitcoin scenario.

[5] G. Cheuque Cerda et al., “Bitcoin price prediction through opinion mining”. The Bitcoin protocol and its underlying crypto currency have started to shape the way we view digital currency, and opened up a large list of new and interesting challenges. Amongst them, we focus on the question of how the price of digital currencies is affected, which is a natural question especially when considering the price rollercoaster we witnessed for bitcoin in 2017-2018. We work under the hypothesis that price is affected by the web footprint of influential people, we refer to them as crypto-influencers. In this paper we provide neural models for predicting bitcoin price. We compare what happens when the model is fed only with recent price history versus what happens when fed, in addition, with a measure of the positivity or negativity of the sayings of these influencers, measured through a sentiment analysis of their twitter posts. We show preliminary evidence that twitter data should indeed help to predict the price of bitcoin, even though the measures we use in this paper have a lot of room for refinement. In particular, we also discuss the challenges of measuring the correct sensation of these posts, and discuss the work that should help improving our discoveries even further.

**CHAPTER II**

**METHODOLOGY**

**2.1 METHOLOGY OF OUR SYSTEM**

The methodology in our project describes the methods as well as the names of the system's algorithms. This section delves into the techniques and methods used in this project. There are primarily two approaches for predicting Bit coin or doge coin, using machine learning classifiers such as LSTM and Monte Carlo simulation of the system.

**2.2 ARTIFICIAL INTELLIGENCE**

Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make computers "smart". As machines become increasingly capable, mental facilities once thought to require intelligence are removed from the definition. AI is an area of computer sciences that emphasizes the creation of intelligent machines that work and reacts like humans. Some of the activities computers with artificial intelligence are designed for include: Face recognition, Learning, Planning, Decision making etc., Artificial intelligence is the use of computer science programming to imitate human thought and action by analyzing data and surroundings, solving or anticipating problems and learning or self-teaching to adapt to a variety of tasks. Artificial intelligence is the science and technology which making machines those are intelligent, particularly smart computer programs as indicated by the scientist John McCarthy. It is a methodology of building a machine, a modified machine controlled robot, and programming thinks keenly, in the comparative way the knowledge people being think. Man-made intelligence is structured by concentrating how the human cerebrum thinks and how people learn, choose, and working while attempting to solving other issue, and afterward utilizing the results of this examination as a premise of creating intelligent programming frameworks.

Artificial intelligence is the process of simulating human intelligence into a programmed machine to think like a human being for solving problems. Artificial Intelligence is a branch of computer science used to develop an intelligent machine that can think and act to solve complex problems rationally like a human being. It is powered by the technique of machine learning, deep learning, heuristic searching, rule based system, Artificial Neural network, Support Vector Machine, Natural Language Processing, etc. There are two types of AI. Narrow AI or weak AI is an artificial intelligence technique that operates within a limited situation and is a human intelligence simulation. Narrow AI is always focused on performing a single task efficiently and effectively. These machines are intelligent systems that are operating under many constraints and restrictions even better than basic human intelligence. Strong AI or Artificial General Intelligence (AGI) is the type of artificial intelligence used in robotics, movies etc. It is a machine with general intelligence that can think and work like a human being; it can apply that intelligence to solve any problem. Artificial Intelligent can be used in various fields such as Smart assistant system like Alexa, Siri, and Cortana these are intelligent voice recognition system helps to provide knowledge as well as enhance our decision making capability.

**2.2.1 Applications of Artificial intelligence**

* It is used to develop an Optimized and personalized healthcare treatment recommendations system.
* AI-based techniques are used in the advertising and marketing industries toϖ develop conversational bots for providing efficient customer service.
* AI used to develop a Robo-advisors intelligent technique for inverters forϖ providing advice for stock trading.
* An artificial intelligent agent can be implemented in Spam detection and filtersϖ on email.
* It is used to develop social media monitoring tools for tracking sensitive contentϖ or fake news.
* AI-based recommendation system provides appropriate recommendations for aϖ particular Song or TV show for online entertainment applications.
* AI-enabled chat bots can be used to offers an instant response to better customerϖ service. AI applications in education help students for better learning experience throughϖ one to one interaction with the teacher.
* Artificial Intelligent based smart home applications with the power of theϖ Internet of things helps to control all devices in the house. Integration of AI with the shopping industry helps to personalized customerϖ services like automatic notifications, alerts, information about new offers, deals, and services provided to the customer
* In medical and health care field AI is used for symptoms mapping and predicting disease.
* AI used in manufacturing industries for performing complex tasks through droneϖ robots.

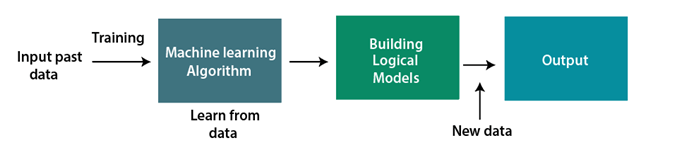
**2.3 MACHINE LEARNING**

Machine learning becoming more and more important because the models are capable of independently adapting to situation when exposing to newer information. They learn from previous computations to produce reliable outputs. This method is a subset of AI which utilizes measurable methods to encourage machine to improve with understanding. This is the kind of programming which gives system the capacity to naturally gain from information without being expressly customized. This implies these projects change their presentation, conduct by gaining from information. Machine learning is connected to theory of computational learning. Machine learning provides the computer with a capability to acquire knowledge without being openly coded. The main objective of the study of learning with machines is, to design algorithms and to develop a thorough analysis of the data. With the use of computers and a proper machine learning strategy, construction of computer models and algorithms for analysis and prediction in the field of data analytics becomes easy. This in turn allows researchers and data analysts to provide reliable decisions and results, and uncover hidden knowledge. Machine learning on a very large scale has become a critical need to handle high dimensional data. The machine should train rapidly, and the ability to train and learn should scale readily with volume and dimension. In healthcare research work, conventional machine learning approaches work efficiently on traditional datasets, but usually their performance deteriorates when they are applied to high dimensional datasets therefore there is need of working with high dimensional data. Machine learning has the capability to acclimate to different situations and to discover and generalize arrangements. This machine learning can be categorized into three categories: Machine learning algorithms are structured into a categorization which based on the anticipated output of an algorithm.

* These algorithm categorizations are: In the supervised learning the algorithm it produces a task which is mapped inputs or input to anticipated outputs. The classification task problem was the ordinary creation for supervised learning.
* In unsupervised learning it make framework of fixed of inputs where considered models are unavailable. In semi supervised learning there is combination of both unlabeled and labeled instances for generating an applicable, suitable class or function. In the reinforcement learning this algorithm studies an approach or strategy or policy which acts on the observation.
* Transduction is similar to supervised learning; change is this it does not openly build a method in its place, efforts to prediction of novel outputs depending on training inputs or new inputs and training outputs.
* These algorithm studies on its individual inductive partiality depending upon the earlier knowledge, considered as knowledge to learn.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms **for building mathematical models and making predictions using historical data or information.** Currently, it is being used for various tasks such as**image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system**, and many more. Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel**in**1959**. We can define it in a summarized way as: “Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed”. A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

**The below block diagram explains the working of Machine Learning algorithm:**



* + 1. **Features of Machine Learning:**
* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data.
  + 1. **Classification of Machine Learning**

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

### 1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering.** Supervised learning can be grouped further in two categories of algorithms:

* **Classification**
* **Regression**

### 2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

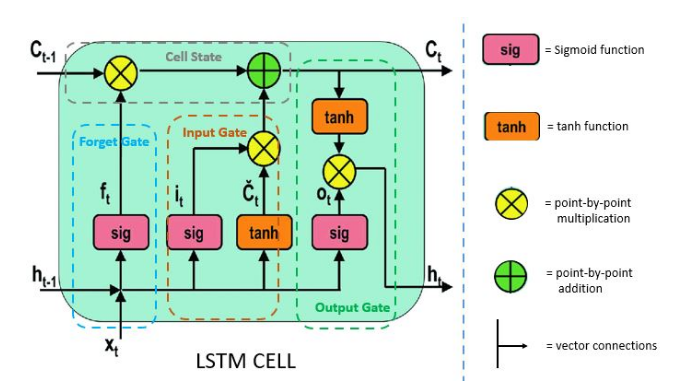
It can be further classifieds into two categories of algorithms:

* **Clustering**
* **Association**
  1. **ALGORITHM USED**

Here, we can use the algorithm are LSTM as well as the Monte Carlo Simulation are used. Both are used to predict the result based on the price of doge coin or bit coin of the system.

* + 1. **LSTM**

LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM stands for long short-term memory networks, used in the field of [Deep Learning](https://intellipaat.com/blog/tutorial/machine-learning-tutorial/introduction-deep-learning/). It is a variety of [recurrent neural networks (RNNs)](https://intellipaat.com/blog/tutorial/artificial-intelligence-tutorial/recurrent-neural-network/) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.



**LSTM Cell**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. LSTMs have three types of gates: input gates, forget gates, and output gates that control the flow of information. The hidden layer output of LSTM includes the hidden state and the memory cell. Only the hidden state is passed into the output layer. The memory cell is entirely internal. LSTM require 4 linear layer (MLP layer) per cell to run at and for each sequence time-step. Linear layers require large amounts of memory bandwidth to be computed, in fact they cannot use many compute unit often because the system has not enough memory bandwidth to feed the computational units. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods. The structure of RNN is very similar to hidden Markov model of the system.

* + 1. **Monte Carlo Simulation**

A Monte Carlo simulation is a type of computational algorithm that estimates the probability of occurrence of an undeterminable event due to the involvement of random variables. The algorithm relies on repeated random sampling in an attempt to determine the probability. This means simulating an event with random inputs a large number of times to obtain your estimation. You can determine other factors as well, and we will see that in the example. Monte Carlo simulations can be utilized in a broad range of fields spanning from economics, gambling, engineering, energy, and anything in-between. So, no matter what career field you are in, it’s an excellent thing to know about. When learning how to build Monte Carlo simulations, it’s best to start with a basic model to understand the fundamentals. The easiest and most common way to do that is with simple games, so we will make use of a dice game in this article. You’ve probably heard the saying, “the house always wins,” so for this example, the house (typically a casino) will have an advantage, and we will show what that means for the player’s possible earnings. Every Monte Carlo simulation will require you to know what your inputs are and what information you are looking to obtain. In our Project the Monte Carlo Simulation works in the time depend series prediction based on the predict the outcome of each and every year of the system.

* 1. **PROGRAMMING TOOLS**

In our project, we can use the Programming tool are Python, Anaconda, Google Colab of the system.

* + 1. **PYTHON**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamically typed language.For example, x=10 .Here, x can be anything such as String, int, etc.Python is an interpreter, object-oriented programming language similar to PERL that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of [operating system](https://whatis.techtarget.com/definition/operating-system-OS)s, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favorite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users.

**Features in Python**

There are many features in Python, some of which are discussed below

* Easy to code
* Free and Open Source
* Object-Oriented Language
* GUI Programming Support
* High-Level Language
* Extensible feature
* Python is Portable language
* Python is Integrated language
* Interpreted Language
  + 1. **ANACONDA**

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) as well as the [conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). The big difference between conda and the [pip package manager](https://en.wikipedia.org/wiki/Pip_(package_manager)) is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail. In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

### 2.5.2.1 Anaconda Navigator

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

The following applications are available by default in Navigator:

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#JupyterLab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* QtConsole
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glue](https://en.wikipedia.org/wiki/Glue_(software))
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [RStudio](https://en.wikipedia.org/wiki/RStudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
  + 1. **GOOGLE COLAB**

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs. Colab resources are not guaranteed and not unlimited, and the usage limits sometimes fluctuate. This is necessary for Colab to be able to provide resources free of charge. Users who are interested in more reliable access to better resources may be interested in [Colab Pro](http://colab.research.google.com/signup?utm_source=faq&utm_medium=link&utm_campaign=seems_too_good). Resources in Colab are prioritized for interactive use cases. We prohibit actions associated with bulk compute, actions that negatively impact others, as well as actions associated with bypassing our policies. The following are disallowed from Colab runtimes:

file hosting, media serving, or other web service offerings not related to interactive compute with Colab

* downloading torrents or engaging in peer-to-peer file-sharing
* using a remote desktop or SSH
* connecting to remote proxies
* mining cryptocurrency
* running denial-of-service attacks
* password cracking
* using multiple accounts to work around access or resource usage restrictions
* creating deepfakes
  + 1. **PACKAGES AND LIBARIES**
* **Numpy:** Numpy is considered as one of the most popular machine learning libraries in python. Array interface is the best and the most important feature of Numpy. This interface can be utilized for expressing images, sound waves, and other binary raw streams as an array of real numbers in N-dimensional.
* **Pandas:** Pandas is a machine learning library in Python that provides high-level data structures and a wide variety of tools for analysis. One of the great features of this library is the ability to translate complex operations with data using one or two commands. Pandas have so many inbuilt methods for grouping, combining data, and filtering, as well as time-series functionality.
* **Matplotlib:** Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.
* **Seaborn:** Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.
* **Scikit-learn:** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

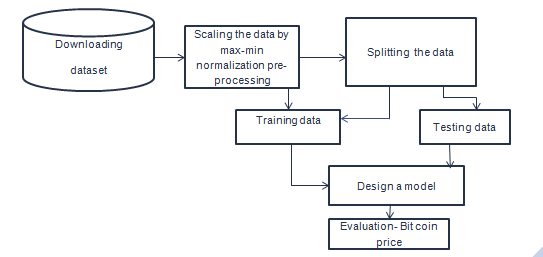
**CHAPTER III**

**SYSTEM DESIGN**

The project methodology is discussed in the previous chapter. This chapter is about system design. The design explains the various modules of the project and demonstrates the overall system architecture of the project.

**3.1 ARCHITECTURE OF OUR PROJECT**

The Figure 1 Shows the System architecture of our method.



**Figure 1 System Architecture of the system**

The Architecture of our Project describes the workflow of the system's proposed method. First, the data is collected in the form of CSV files, which are then pre-processed in order to scale, normalize, and clean the unwanted data in our system. The data is then visualized using the system's EDA concept. Following that, the data is divided into two parts: training and testing of the system. To train the data, we can use two algorithms: LSTM and Monte Carlo simulation. We want to test the data based on the system's prediction of bit coin or doge coin once the training phase is complete. Finally, the trained data are tested to demonstrate the system's output prediction. Following that, the result based on the system's accuracy model will be displayed.

**3.2 USE CASE DIAGRAM**

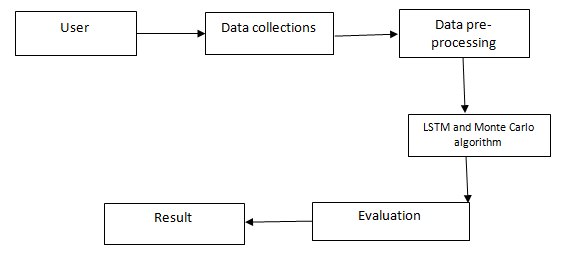
A Figure 2 Shows the Use Case Diagram of the System.

**Use Case Diagram**

User

**3.3 COLLABORATIVE DIAGRAM**

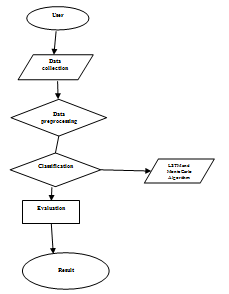
A Figure 3 Shows the Collaborative Diagram of the System.

****

**Collaborative Diagram**

**3.4 DATA FLOW DIAGRAM**

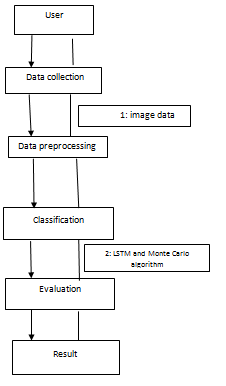
A Figure 4 Shows the Data Flow Diagram of the System.



**Data Flow Diagram**

**3.5 ER DIAGRAM**

A Figure 5 Shows the ER Diagram of the System.

****

**ER Diagram**

**CHAPTER IV**

**DEVELOPMENT PROCESS**

In the development process, we can describe the model are datasets collection, data preprocessing, data visualization, Training and testing the data and Prediction of the result of the system.

**4.1 DATASETS COLLECTION**

In this project, we have used the dataset is doge coin Historical Data and it is available in Kaggle. The dataset has seven columns: Year, open, high, low, close, volume and Adjust volume. The dataset happens to have many null values, so before proceeding further, first the rows with null values were dropped for better prediction of the data and finally a set of data without any null value is obtained. Dogecoin dataset collected live and stored as bitcoin.csv file is considered as dataset, which is split into train set and test set. We have considered 80% of data as training input for our machine learning algorithm model to train the model. The remaining 20% of data is considered as test for result prediction. We exploited decision are LSTM and Monte Carlo Simulation are used to predict the price trend for 20% test input and the predicted values to plotted and show the value of the accuracy of the system.

* 1. **DATA PREPROCESSING**

After cleaning the dataset, it was prepared for fitting it into machine learning algorithms. At first, two more columns were removed that is weighted price and timestamp for it to be fitted to the algorithms. Then the data was scaled for better fitting with Min Max Scaler.. Before exploring and visualizing the data, it is necessary to clean the dataset available. Cleaning includes getting rid of the null values or irrelevant zero values are removed of the system. Then read the data and load it into the pandas library the data can be cleaned . Since bitcoin data is basically time-series data, the first step is to perform some data exploratory techniques before forecasting. That includes checking whether the time-series is stationary. This is performed by a seasonal decomposition of the data to estimate its trend and seasonality. In the case of non- stationary, it can perform techniques to stationeries the series. Finally to clean the unwanted data to show the result in terms of efficient of the system.

* 1. **DATA VISUALIZATION**

Using the EDA Concept, we can visualize the data based on the open price, close price, high price, and low price of our system's data. In addition, to show the data in the distribution plot of our data to show the system's result.

* 1. **TRAIN THE TESTING THE MODEL**

By using LSTM and Monte Carlo simulation algorithm to train the model of the system.

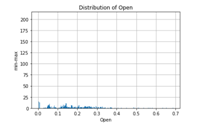
* 1. **PREDICTION**

LSTM and MCS are very powerful in sequence prediction problems because they’re able to store past information. This is important in case of predicting the future bitcoin prices. The pre-processed data is provided as the input to the model and then compile the model after the compilation of the model the evaluation is done. The prediction is visualized by plotting the graph with importing the matplotlib library. Two hidden LSTM layers were chosen. For a time series task two layers is enough to find non-linear relationships among the data for a time series task. Three and four layers were tested but didn’t improve validation performance. Several layers can be required for tasks such as image recognition when the number of features is significantly large. Finally to show the results in terms of the Accuracy pattern of the system. Accuracy is represented as the ratio between the sum of predicted values and the sum of original values multiplied by 100. The loss we have got in terms of mean squared error. We are representing the accuracy and loss comparisons through graphs and histograms. The above graphs help us about how well fitted the predicted data is with respect to the testing data.

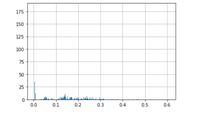
**CHAPTER V**

**RESULTS AND DISCUSSION**

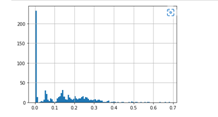
In the results and discussion part, we discussed the output Results of our project.



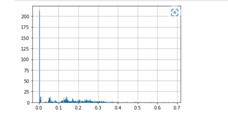
**Distribution Plot of Open**

****

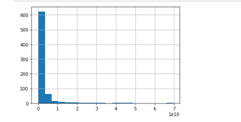
**Distribution plot of Low**

****

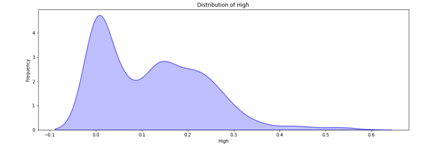
**Distribution plot of Close**

****

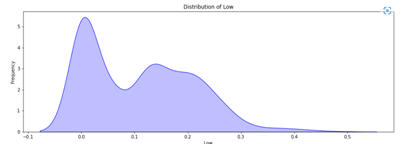
**Distribution plot of AdjClose**

****

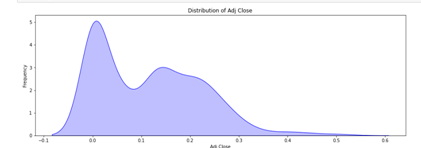
**Distribution plot of Volume**

****

**Frequency Vs Distribution Plot of High**

****

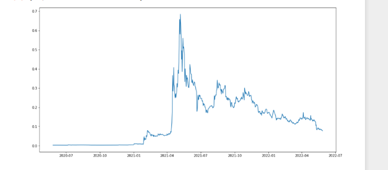
**Frequency Vs Distribution Plot of Low**

****

**Frequency Vs Distribution Plot of Adjclose**

****

**Frequency Vs Distribution Plot of Total Volume**

****

**Results of LSTM**

****

**Results of Monte Carlo Simulation**

**CHAPTER VI**

**CONCLUSION**

The main purpose of this project is to predict the price of the bitcoin with the use of Machine learning algorithms such as long short term memory LSTM and Monte Carlo Simulation of the system It helps in the trading and investment of the money. LSTM results into the least RMSE value when bitcoin parameters are considered for price prediction when compared to the other algorithms. The proposed approach helps the researcher to determine optimal input variables for building LSTM time series forecasting model without trial and-error process. Finally, the prediction of result shows the better performance in terms of accuracy of our system.

**REFERENCES**

[1] Greaves, A., & Au, B. (2021). Using the bitcoin transaction graph to predict the price of bitcoin. No Data.

[2] Hayes, A. S. (2020). Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin. Telematics and Informatics, 34(7), 1308-1321.

[3] Shah, D., & Zhang, K. (2014, September). Bayesian regression and Bitcoin. In Communication, Control, and Computing (Allerton), 2021 52nd Annual Allerton Conference on (pp. 409-414). IEEE.

[4] Indra N I, Yassin I M, Zabidi A, Rizman Z I. Non-linear autoregressive with exogenous input (mrx) bitcoin price prediction model using so-optimized parameters and moving average technical indicators. J. Fundam. Appl. Sci., 2021, 9(3S), 791-808

[5] G. Cheuque Cerda and J. L. Reutter, ``Bitcoin price prediction through opinion mining,'' in Proc. World Wide Web Conf., New York, NY, USA, 2019, pp. 755\_762.

[6] G. Sera\_ni, P. Yi, Q. Zhang, M. Brambilla, J. Wang, Y. Hu, and B. Li, `Sentiment-driven price prediction of the bitcoin based on statistical and deep learning approaches,'' in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2020, pp. 1\_8.

[7] T. Phaladisailoed and T. Numnonda, ``Machine learning models comparison for bitcoin price prediction,'' in Proc. 10th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE), Jul. 2018, pp. 506\_511.

[8] S. Khuntia and J. Pattanayak, ``Adaptive market hypothesis and evolving predictability of bitcoin,'' Econ. Lett., vol. 167, pp. 26\_28, Dec. 2021.

[9] G. Giudici, A. Milne, and D. Vinogradov, ``Cryptocurrencies: Market analysis and perspectives,'' J. Ind. Bus. Econ., vol. 47, no. 1, pp. 1\_18, Mar. 2020.

[10] M. Watch. Cryptocurrency Market Grow With a high Cagr-Global Industry Analysis, Key Manufacturers, Trends, Size, and Forecasts During 2025. Accessed: Mar. 1, 2021.

[11] K. Sheth, K. Patel, H. Shah, S. Tanwar, R. Gupta, and N. Kumar, ``A taxonomy of ai techniques for 6G communication networks,'' Com- put. Commun., vol. 161, pp. 279\_303, Jun. 2020.

[12] Y. Sovbetov, ``Factors in\_uencing cryptocurrency prices: Evidence from Bitcoin, Ethereum, dash, Litcoin, and Monero,'' J. Econ. Financial Anal., vol. 2, no. 2, pp. 1\_27, 2018.

[13] O. Angela andY. Sun, ``Factors affecting cryptocurrency prices: Evidence from ethereum,'' in Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech),

Aug. 2020, pp. 318\_323.

[14] A. G. Titan, ``The ef\_cient market hypothesis: Reviewof specialized literature and empirical research,'' Proc. Econ. Finance, vol. 32, pp. 442\_449, Sep. 2015.

[15] A. Thakkar and K. Chaudhari, ``Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions,'' Inf. Fusion, vol. 65, pp. 95\_107, Sep. 2021.

**APPENDIX**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings("ignore")

data=pd.read\_csv(r"/content/data.csv")

data

data.columns

data.shape

data.info()

data.describe()

data.isnull().sum()

data.nunique()

data['Open'].hist(bins=400)

plt.xlabel("Open")

plt.ylabel("min-max")

plt.title("Distribution of Open")

plt.show()

data['High'].hist(bins=250)

plt.xlabel("Open")

plt.ylabel("min-max")

plt.title("Distribution of Open")

plt.show()

data['Low'].hist(bins=550)

plt.show()

data['Close'].hist(bins=100)

plt.show()

data['Adj Close'].hist(bins=300)

plt.show()

data['Volume'].hist(bins=20)

plt.show()

#Distribution plot

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Open']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Open")

plt.ylabel("Frequency")

plt.title("Distribution of Open")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['High']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("High")

plt.ylabel("Frequency")

plt.title("Distribution of High")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Low']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Low")

plt.ylabel("Frequency")

plt.title("Distribution of Low")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Close']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Close")

plt.ylabel("Frequency")

plt.title("Distribution of Last")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Adj Close']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Adj Close")

plt.ylabel("Frequency")

plt.title("Distribution of Adj Close")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Volume']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Total Volume")

plt.ylabel("Frequency")

plt.title("Distribution of Total Volume")

plt.show()

plt.figure(figsize=(15,5))

sns.distplot(np.log1p(data['Low']), hist=False, color="b", kde\_kws={"shade": True})

plt.xlabel("Low")

plt.ylabel("Frequency")

plt.title("Distribution of Low")

plt.show()

#setting index as date

data['Date'] = pd.to\_datetime(data.Date,format='%d-%m-%Y')

data.index = data['Date']

#plot

plt.figure(figsize=(16,8))

plt.plot(data['Date'], label='Date wise Crypto Mining')

#setting index as date

data['Date'] = pd.to\_datetime(data.Date,format='%Y-%m-%d')

data.index = data['Date']

#plot

plt.figure(figsize=(16,8))

plt.plot(data['Close'], label='Close Volume')

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import Dense, Dropout, LSTM

data1 = data.sort\_index(ascending=True, axis=0)

new\_data = pd.DataFrame(index=range(0,len(data)),columns=['Date', 'Close'])

for i in range(0,len(data1)):

    new\_data['Date'][i] = data1['Date'][i]

    new\_data['Close'][i] = data1['Close'][i]

#setting index

new\_data.index = new\_data.Date

new\_data.drop('Date', axis=1, inplace=True)

#split the train and test sets

dataset = new\_data.values

train = dataset[0:987,:]

valid = dataset[400:,:]

train

valid

#converting dataset into x\_train and y\_train

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_data = scaler.fit\_transform(dataset)

x\_train, y\_train = [], []

for i in range(60,len(train)):

    x\_train.append(scaled\_data[i-60:i,0])

    y\_train.append(scaled\_data[i,0])

x\_train, y\_train = np.array(x\_train), np.array(y\_train)

x\_train = np.reshape(x\_train, (x\_train.shape[0],x\_train.shape[1],1))

x\_train

import numpy as numpy

import cv2

import tensorflow

from tensorflow.keras import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D

from keras.layers import MaxPool2D

from keras.preprocessing.image import ImageDataGenerator

# create and fit the LSTM network

model = Sequential()

model.add(LSTM(units=50, return\_sequences=True, input\_shape=(x\_train.shape[1],1)))

model.add(LSTM(units=50))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error', optimizer='adam',metrics=['mse'])

model.fit(x\_train, y\_train, epochs=15, batch\_size=25, verbose=2)

model.summary()

#predicting 246 values, using past 60 from the train data

inputs = new\_data[len(new\_data) - len(valid) - 60:].values

inputs = inputs.reshape(-1,1)

inputs  = scaler.transform(inputs)

inputs.shape

x\_test = []

for i in range(59,inputs.shape[0]):

    x\_test.append(inputs[i-59:i,0])

x\_test = np.array(x\_test)

x\_test = np.reshape(x\_test, (x\_test.shape[0],x\_test.shape[1],1))

closing\_price = model.predict(x\_test)

closing\_price = scaler.inverse\_transform(closing\_price)

closing\_price=closing\_price\*100

print(closing\_price)

score, acc = model.evaluate(x\_test,x\_test, batch\_size=25, verbose=2)

print('score', score)

print('accuracy', acc)

from tensorflow.keras.metrics import RootMeanSquaredError

model = Sequential()

model.add(LSTM(units=50, return\_sequences=True, input\_shape=(x\_train.shape[1],1)))

model.add(LSTM(units=50))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error', optimizer='adam',metrics=RootMeanSquaredError())

model.fit(x\_train, y\_train, epochs=15, batch\_size=25, verbose=2)

model.summary()

x\_test = []

for i in range(59,inputs.shape[0]):

x\_test.append(inputs[i-59:i,0])

x\_test = np.array(x\_test)

x\_test = np.reshape(x\_test, (x\_test.shape[0],x\_test.shape[1],1))

closing\_price = model.predict(x\_test)

closing\_price = scaler.inverse\_transform(closing\_price)

closing\_price=closing\_price\*100

print(closing\_price)

score, acc = model.evaluate(x\_test,x\_test, batch\_size=25, verbose=2)

print('score', score)

print('accuracy', acc)

train = new\_data[:987]

valid = new\_data[987:]

import pandas\_datareader.data as web

import pandas as pd

from datetime import datetime

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import style

startdate = '2017-01-01'

today = datetime.today().strftime('%Y-%m-%d')

startdate, today

price = web.DataReader('AAPL', data\_source = 'yahoo', start = startdate, end = today)['Adj Close']

price

daily return and volatiltiy

daily\_return = price.pct\_change()

daily\_vol = daily\_return.std()

daily\_return, daily\_vol

last\_price = price[-1]

last\_price

# Monte Carlo Simulation setup

num\_simulation = 5

num\_days = 252

simulation\_df = pd.DataFrame()

for x in range(num\_simulation):

count = 0

daily\_vol = daily\_return.std()

price\_series = []

prices = last\_price\*(1 + np.random.normal(0, daily\_vol))

price\_series.append(prices)

for y in range(num\_days):

if count == 251:

break

prices = price\_series[count]\*(1+np.random.normal(0, daily\_vol))

price\_series.append(prices)

count += 1

simulation\_df[x] = price\_series

simulation\_df

fig = plt.figure()

fig.suptitle('Monte Carlo Simulation')

plt.plot(simulation\_df, lw = 1)

plt.axhline(y = last\_price, color = 'r', linestyle = '-')

plt.xlabel('Day')

plt.ylabel('Price')

plt.show()