**ABSTRACT**

A Cryptocurrency is a digital currency designed to work as a medium of exchange through a computer network that is not reliant on any central authority such as a government or bank, to uphold or maintain it.The dynamic and highly volatile nature of cryptocurrency markets presents unique challenges and opportunities for investors and analysts.

This project aims to conduct a comprehensive time series analysis of cryptocurrency prices, focusing on major digital assets such as Bitcoin, Ethereum, Maker, Zcash and Litecoin. The primary objectives are to identify underlying patterns, seasonal trends, and volatility structures, and to develop robust predictive models for future price movements. Key findings from the time series analysis reveal distinct price patterns, including periods of high and low volatility. The project also forecasts future price movements of cryptocurrency prices based on the previous year’s data using forecasting techniques. Building upon the insights gained from EDA (Exploratory Data Analysis), various forecasting models are employed including time series models such as ARIMA (Auto Regressive Integrated Moving Average), deep learning techniques such as Long Short-Term Memory (LSTM) networks. These models are trained and tested using historical price data to predict future price movements with accuracy.

The study concludes with recommendations for future research directions to further refine forecasting techniques and improve understanding of cryptocurrency price dynamics.

**Chapter 1**

**INTRODUCTION**

**1.1 WHAT IS CRYPTOCURRENCY?**

Cryptocurrency, also known as Crypto, is a digital currency that operates on a computer network without relying on a central authority like government or bank for its upkeep. Cryptocurrencies are digital or virtual currencies protected by cryptography, making counterfeiting and double spending nearly impossible. They are typically decentralized, blockchain-based networks operated by dispersed computing networks. They are not distributed by a central agency, making them potentially resistant to government intervention or abuse. The other aspect is that it is powered by block chain technology, which is incredibly complicated and aims to store data in such a way that it is difficult or impossible to alter, hack, or cheat the system. Cryptocurrency is produced by a collective system at a rate defined by the system’s creation and public statement. In contrast to centralized banking systems like the US Federal Reserve System, cryptocurrency do not require corporate boards or government to produce new units and have not provided backing for other firms, banks, or corporate entities holding asset value.

David Chaum, an American cryptographer, introduced ecash in 1983 and implemented it through Digicash in 1995. Digicash required user software to withdraw notes from a bank and designate encrypted keys, making the digital currency untraceable. In 1996, the National Security Agency published a paper describing a cryptocurrency system, “How to Make Mint: The Cryptography of Anonymous Electronic Cash”. In 1998, Wei Dai described “b-money”, an anonymous. Distributed electronic cash system, and Nick Szabo described BitGold, an electronic currency system that required users to complete a proof of work function with cryptographically put together and published solutions.



**Fig 1.1: Cryptocurrencies**

**1.2 JOURNEY OF CRYPTOCURRENCIES IN INDIA**

India’s investment in cryptocurrencies, particularly Bitcoin, has seen a significant increase since 2020, with over 1.5-2 crore Indians investing in the asset class, reaching $10 billion in November. This shift in investment paradigm, suggest a shift from the country’s preference for gold and other safer assets. The Cryptocurrency and Regulation of Official Digital Currency Bill is set to be introduced highlighting the virtual asset’s journey thus far.



**Fig 1.2: Cryptocurrencies in India**

**2008: Inception of Cryptocurrencies**

Santoshi Nakamoto, a pseudonymous developer, published a paper in 2008 titles “Bitcoin: A Peer-to-Peer Electronic Cash System” marking the beginning of cryptocurrency’s journey.

**2010: First Sale Using Crypto**

Two years later, Bitcoin’s first sale occurred with a swap of 10,000 Bitcoin for two pizzas, establishing a cash value for cryptocurrencies. Other cryptocurrencies like Litecoin, Namecoin and Swiftcoin emerged, gaining traction and influencing the digital asset.

**2013: RBI Issues First Circular Regarding Cryptocurrencies**

In 2013, the Reserve Bank of India issued a circular warning users of potential security risks associated with the use of virtual currencies, as crypto investments in India surged and exchanges like Zebpay and Pocket Bits emerged.

**2016-2018: Demonetisation and RBI’s Banking Ban on Crypto**

The demonetisation experiment in India led to increased preference for digital payments, which subsequently boosted crypto investments. Indian banks continued to allow transactions on cryptocurrency exchanges, prompting the RBI to expressing concerns about virtual coins. By the end of 2017, a warning clarified that virtual currencies are not legal tenders. In March 2018, the Central Board of Digital Tax submitted a draft scheme for banning virtual currencies, which was later restraining banks, NBFCs and payment system providers from dealing with virtual currencies and providing services to exchanges. This resulted in 99% drop in trading volumes.

**2020: Supreme Court Stikes Down the Crypto Banking Ban**

The RBI circular ban on cryptocurrency led to a writ petition in the Supreme Court, which ultimately stricken the ban, declaring it unconstitutional. This ruling coincided with the crypto boom, allowing cryptocurrency exchanges to regain their momentum.

**2021: Announcement of Crypto Bill**

The Indian government plans to introduce a bill to create a sovereign digital currency and ban private cryptocurrencies. In November 2021, the Standing Committee on Finance met with the Blockchain and Crypto Assets Council (BACC) and other cryptocurrency representatives, concluding that cryptocurrencies should be regulated, not banned. Prime Minister Narendra Modi chaired a meeting on cryptocurrencies in December 2021.

**1.3 LIST OF CRYPTOCURRENCIES**

As of March 2024, there are 13,217 cryptocurrencies in existence. However, not all cryptocurrencies are active or valuable. Discounting many “dead” cryptos leaves only around 8,985 active cryptocurrencies.

**1.3.1 BITCOIN**



**Fig 1.3.1: Bitcoin Logo**

Bitcoin (Abbreviation: ***BTC***) is the first decentralized cryptocurrency. Bitcoin is an innovative payment network and new kind of money. It uses peer-to-peer technology to operate with no central authority or banks; managing transactions and the issuing of bitcoins is carried out collectively by the network. Bitcoin is open-source; its design is public nobody owns or controls Bitcoin and everyone can take part.

Bitcoins, invented in 2008 by Santoshi Nakamoto, is a free-market currency that began as an open-source implementation in 2009. It was adopted as legal tender in EI Salvador in 2021. Currently, it is primarily used as a store of value and is considered an economic bubble by scholars. Due to its pseudonymous nature, its use by criminals has led to its bank by several countries as of 2021.

Bitcoin’s decentralization of money has its roots in Austrian economics, particularly Friedrich von Hayek’s book. The Denationalization of Money. Sociologist Nigel Dodd argues that the essence of bitcoin is to remove money from social and governmental control. The Economist described bitcoin as a techno-anarchist project to create an online version of cash, attracting libertarians and anarchists. Economist Paul Krugman argues that cryptocurrencies like Bitcoin are only used by bank skeptics and criminals.

Digital cash technologies began with David Chaum’s ecash in the 198-s, followed by Cynthia Dwork and Moni Naor’s 1992 idea of computational puzzles having value. Adam Back developed Hashcash in 1997, and Cypherpunks Wei Dai and Nick Szabo proposed distributed digital scarcity, based cryptocurrencies in 1998 and 2004 respectively. However, these attempts were unsuccessful due to centralized control, lack of protection against double-spending and resistance to Sybil attacks.



**Fig 1.3.2: Bitcoin**

**2008-2009: Creation**

Bitcoin.org was registered on 18 August,2008. Satoshi Nakamoto’s white paper, “Bitcoin: A Peer-to-Peer Electronic Cash System”, was published on 31 October,2008. Nakamoto implemented bitcoin software as open-source code and released it in January 2009. The first decentralized, Sybil resistant, Byzantine fault tolerant digital cash system, known as the first blockchain, was created on 3 January 2009. The first known commercial transaction using bitcoin occurred on 22 May 2010, when the programmer Laszlo Hanyecz bought two Papa John’s pizzas for ₿ 10,000, in what would late be celebrated as “Bitcoin Pizza Day”.

**2010-2012: Early Growth**

Block chain analyst Nakamoto mined around one million bitcoins before disappearing in 2010. He handed the code repository to Gavin Andresen, who late became lead developer at the Bitcoin Foundation. Bitcoin’s first major users were black markets like Silk Road, which exclusively accepted bitcoins as payment for 30 months, worth around $214 million.

**2013-2014: First regulatory actions**

In 2013, the US Financial Crimes Enforcement Network (FinCEN) established guidelines for decentralized virtual currencies like bitcoin, classifying American miners as money services businesses. In 2013, authorities seized Mt. Gox, ₿11,02 from a man attempting to buy illegal substances, and ₿30,000 from Silk Road. In December 2013, the People’s Bank of China prohibited Chinese financial institutions from using bitcoin, leading to a drop in bitcoin value and Baidu no longer accepting bitcoins for certain services.

**2015-2019**

In 2017, the University of Cambridge estimated that 2.9-5.8 million unique users used cryptocurrency wallets, primarily using bitcoin. SegWit software upgrade in August 2017 improved scalability and support for the Lightning Network. Bitcoin Cash was created by SegWit opponents. The Chicago Mercantile Exchange introduced bitcoin futures in December 2017. However, Bitcoin’s price crashed in February 2018 due to China’s ban on trading and hacks from exchanges.

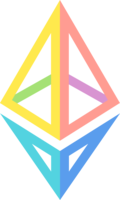
**2020-Present**

In 2020, major companies and institutions like MicroStrategy, Square, and Mass mutual invested $250 million, $50 million, and $100 million in bitcoin, respectively and PayPal added US support in November 2020.

Bitcoin’s market capitalization reached $1 trillion in February 2021, marking the first time in reached $1 trillion. In November, the Taproot soft-fork upgrade improved smart contract functionally and Lightning Network support. In September 2021, Bitcoin became legal tender in EI Salvador alongside the US dollar. In October 2021, the first bitcoin futures Exchange-traded fund (ETF), called BITO, from ProShares was approved by the SEC and listed on the CME.

Bitcoin price fell in May and June 2022 due to TerraUSD collapse and Celsius Network’s collapse. In 2023, non-fungible tokens (NFTs)-on Bitcoin, went live and the first 11 US spot bitcoin ETFs began trading in January 2024. As of June 2023, River Financial estimated the Bitcoin had 81.7 million users, about 1% of the global population.

**1.3.2 ETHEREUM**



**Fig 1.3.3: Ethereum Logo**

Ethereum (Abbreviation: ***ETH***) is the community-run technology powering the cryptocurrency ether (ETH) and thousands of decentralized applications. Ethereum, founded in 2013 by programmer Vitalik Buterin and other founders, is a decentralized blockchain that enables the deployment of permanent, imputable applications for decentralized finance. It allows users to create and exchange non-fungible tokens (NFTs) tied to unique digital assets, such as images. The network went live on 30 July 2015, and many other cryptocurrencies use the ERC-20 token standard for initial coin offerings. On 15 September 2022, Ethereum transitioned its consensus mechanism from proof-of work (PoW) to proof-of-stake (PoS) through the Merge process, reducing Ethereum’s energy usage by 99%.

Ethereum 2.0 (ETH2) was a series of upgrades, known as “phases”, designed to transition the network’s consensus mechanism to proof-of-stake increase transaction throughput through execution sharding, and enhance EVM architecture. On March 13, 2024, the “Dencun” or “Deneb-Cancun” upgrade, a second phase of the Ethereum blockchain upgrade, was launched, reducing transaction fees on Layer 2 networks.



**Fig 1.3.4: Ethereum**

**2013-2014: Founding**

Ethereum was initially described in late 2013 in a white paper by Vitalik Buterin, a programmer and co-founder of Bitcoin Magazine, that described a way to build decentralized application. He argues that blockchain technology could benefit from other applications besides money and needed a more robust language for application development. He briefly worked with eTora CEO Yoni Assia on the coloured coins project, but after disagreement, he proposed the development of a new platform with a Turing-complete programming language, which would eventually become Ethereum. This led to the attachment of real-world assets to the blockchain.

Ethereum was announced at the North American Bitcoin Conference in Miami in January 2014. The founders, Gavin Wood, Charles Hoskinson, and Anthony Di Iorio, rented a house in Miami to develop their vision. Di Iorio invited Joseph Lubin and Morgen Peck to witness the experience. Six months later, the founders met in Switzerland, where Buterin announced the project would proceed as a non-profit. Hoskinson left the project and founded IOHK, a blockchain company responsible for Cardano.

**2014: Development**

Ethereum, a blockchain-based digital currency, was developed in 2014 by Swiss company Ethereum Switzerland GmbH. Gavin Wood, then the CEO, specified the Ethereum Virtual Machine in the Ethereum Yellow Paper. The Ethereum Foundation was established, funded by an online public crowd sale from July to August 2014. Although Ethereum was praised for its technical innovations, concerns about its security and scalability were raised. The development was funded by an online public crowd sale, which involved buying Ethereum value tokens with bitcoin.

**2014-2016: Launch and the DAO event**

The Ethereum Foundation developed several prototypes of Ethereum over 18 months in 2014 and 2015, with "Olympic" being the last prototype and public beta pre-release. Users received a bug bounty of 25,000 ether for stress-testing the blockchain. On July 30, 2015, "Frontier" marked the official launch of the Ethereum platform, with its "genesis block" containing 8,893 transactions and a block reward of 5 ETH.

Ethereum has undergone several protocol upgrades since its launch, affecting its functionality and incentive structures. These upgrades are done through a hard fork. In 2016, The DAO raised $150 million in a crowd sale to fund the project. However, in June 2016, $50 million of DAO tokens were stolen by an unknown hacker, sparking a debate on whether Ethereum should perform a hard fork to reappropriate the funds. The fork resulted in the network splitting into two blockchains: Ethereum with the theft reversed and Ethereum Classic, which continued on the original chain.

**2017-present: Continued development and milestones**

In March 2017, the Enterprise Ethereum Alliance (EEA) was formed by blockchain startups, research groups, and Fortune 500 companies. By May 2017, the nonprofit organization had 116 enterprise members, including ConsenSys, CME Group, Cornell University's research group, Toyota Research Institute, Samsung SDS, Microsoft, Intel, J. P. Morgan, Cooley LLP, Merck KGaA, DTCC, Deloitte, Accenture, Banco Santander, BNY Mellon, ING, and National Bank of Canada.

**1.3.3 LITECOIN**



**Fig 1.3.5: Litecoin Logo**

Litecoin (Abbreviation: ***LTC***) is a peer-to-peer Internet currency that enables instant, near-zero cost payments to anyone in the world. Litecoin is an opensource, global payment network that is fully decentralized without any central authorities. Mathematics secures the network and empowers individuals to control their own finances. Litecoin features faster transaction confirmation times and improved storage efficiency than the leading math-based currency. With substantial industry support, trade volume and liquidity. Litecoin is a proven medium of commerce complementary to Bitcoin. Inspired by Bitcoin, Litecoin was among the earliest altcoins, starting in October 2011.

Litecoin is different in some ways from Bitcoin:

* Targeted block time is every 2.5 minutes, faster than Bitcoin’s 10 minutes.
* Scrypt, an alternative proof-of-work algorithm, prevents use of ASICs.
* Litecoin is mined with Dogecoin.
* Maximum circulating supply is Ł84,000,000, four times larger than Bitcoin’s ₿21,000,000.
* MWEB optional privacy added in May 2022 for private wallet and transaction amounts.
* Third-party vendors like BitPay provide Litecoin’s point of sale infrastructure.



**Fig 1.3.6: Litecoin**

**Pre-Litecoin:**

In 2011, Bitcoin mining was primarily performed by GPUs, leading to concerns about high barriers to entry and the diminishing value of CPU resources. Tenebrix (TBX), a new alternative currency, was created using Bitcoin code and replacing the SHA-256 rounds with the scrypt function, designed to be expensive to accelerate with FPGA or ASIC chips. Tenebrix was "GPU-resistant" and could utilize available CPU resources from bitcoin miners. However, the developers included a clause allowing them to claim 7.7 million TBX at no cost, which was criticized by users.

Charlie Lee, a Google employee and later Coinbase engineering director, developed Fairbrix (FBX), an alternative version of Tenebrix, which combines Litecoin's scrypt mining algorithm with Bitcoin's limited money supply and other modifications.

**Creation and Launch:**

Lee released Litecoin on October 7, 2011, as an open-source client on GitHub. It was a source code fork of Bitcoin Core, with a reduced block generation time, increased coin maximum, a different hashing algorithm (scrypt), faster difficulty retargets, and a modified GUI.

**2011-2016:**

Litecoin's early growth was facilitated by increased exchange availability and liquidity on early exchanges like BTC-e. In November 2013, the aggregate value of Litecoin experienced a 100% leap within 24 hours. In 2014, Lee suggested merging Dogecoin with Litecoin, and in September 2014, Dogecoin began merge-mining with Litecoin.

**2017-2021:**

PayPal introduced a Crypto feature in 2020, allowing users to buy a derivative of Litecoin, along with Bitcoin, Ethereum, and Bitcoin Cash. However, in September 2021, a fake press release announcing a partnership between Litecoin and Walmart caused a 30% increase in Litecoin's price before being revealed as a hoax.

**2022-present:**

In May 2022, the Litecoin network activated the MWEB (Mimblewimble Extension Blocks) upgrade, allowing users to send confidential transactions. In June 2022, PayPal added the ability to transfer Litecoin, along with Bitcoin, Ethereum, and Bitcoin Cash, between PayPal and other wallets and exchanges.

**1.3.4 ZCASH**



**Fig 1.3.7: Zcash Logo**

Zcash (Abbreviation: ***ZEC***) is a cryptocurrency built on a decentralized blockchain that uses advanced applied cryptography to provide enhanced security and privacy through shielded addresses. It was originally built on an open-source code similar to Bitcoin.

Zcash offers private transactors the option of “selective disclosure”, allowing a user to prove payment for auditing purposed. One such reason is to make it easier for private transactions to comply with anti-money laundering laws and tax regulations.

Zcash transactions can be transparent, controlled by a "t-addr" or shielded by a "z-addr". A shielded transaction uses a non-interactive zero-knowledge proof called "zk-SNARK" to provide anonymity to coin holders. As of December 2017, only around 4% of Zcash coins were in the shielded pool, as most cryptocurrency wallet programs and web-based wallets did not support z-addrs. The anonymity set in the shielded pool can be significantly reduced by heuristics-based identifiable usage patterns.

Miners receive 80% block rewards, with 20% going to the Zcash development fund, with 8% to Zcash Open Major Grants, 7% to Electric Coin Co., and 5% to The Zcash Foundation.



**Fig 1.3.8: Zcash**

**2013-present:**

In 2013, Johns Hopkins University professor Matthew Green and graduate students developed Zcash, which was completed by the for-profit Zcash Company, led by Colorado-based Zooko Wilcox. In 2016, the company raised over $3 million from Silicon Valley venture capitalists to complete the project.

Zcash, first mined in October 2016, experienced high demand and quickly became trading at $5,000 per piece. Ten percent of all coins mined for the first four years were allocated to the Zcash Company, employees, investors, and the non-profit Zcash Foundation.

The setup of Zcash required a trusted procedure called "The Ceremony" to create the private key, ensuring privacy and preventing any person or computer from retaining a copy or regenerating the key. The Ceremony was a two-day process executed simultaneously in six global locations, without prior knowledge of who would be participating. The private key was generated and used to instantiate Zcash, and the computers used in the process were reportedly destroyed. In 2022, Edward Snowden claimed to have participated in The Ceremony under a pseudonym.

On February 21, 2019, the “Zcash Company” announced a re-branding as the Electric Coin Company (ECC).

On May 19,2020, a paper titled "Alt-Coin Traceability" examined the privacy of Zcash and Monero, concluding that more academic research is needed and that Zcash's privacy guarantees are questionable. The paper argued that current heuristics from a 2018 Usenix Security Symposium paper make Zcash less anonymous and more traceable, highlighting the need for further research.

On June 8, 2020, Chainalysis has added support for Zcash to its Chainalysis Reactor and "Know Your Transaction" technologies, enabling it to trace and provide transaction values and sender/receiver addresses for over 99% of Zcash activity. This is achieved due to most Zcash users not using privacy-enhancing features. Chainalysis also cites a RAND corporation research report revealing less than 0.2% of cryptocurrency addresses on the dark web were Zcash addresses.

The Electronic Coin Company announced on October 12, 2020, that a majority of investors and owners of Zerocoin Electric Coin Company LLC (ECC) have agreed to donate the company as the wholly owned property of Bootstrap. ECC's shareholders voted in Favor of donating 100% of the company's shares to Bootstrap, and on March 30, 2021, the company's transparency report stated that it is now a wholly owned entity of the 501(c)3 Bootstrap.

In September 2023, ViaBTC, a mining pool, seized control of 51% of Zcash's hashing power, raising concerns about potential attacks. Coinbase swiftly implemented defensive measures, including putting Zcash markets into "limit-only" mode, to protect users and prevent significant price swings during the situation.

**1.3.5 MAKER**



**Fig 1.3.9: Maker logo**

Maker (Abbreviation: ***MKR***) is the nickname for MakerDAO, which is the decentralized autonomous organization smart contract platform that created the DAI stablecoin and the MKR token, which funds MakerDAO and serves as its governance token.

For all of its vast differences, holding MKR is somewhat similar to owning stock in a traditional company, in the sense that the shareholders have a say in determining how the company functions. The Maker ecosystem was one of the first DeFi projects to achieve significant success – a testament to the effectiveness of truly decentralized governance.

The Maker Protocol creates new Dai through smart contracts called Maker Vaults, which can be created through web UIs and apps like Oasis Borrow or Instadapp. Users must pay back the generated Dai and a stability fee to retrieve their collateralized crypto from the smart contract.

The MKR token governs the Maker Protocol, allowing smart contracts to be deployed by any Ethereum address. The community can vote on proposals, with the Ethereum address receiving the most MKR approvals granted administrative access to make changes to the Maker Protocol.

The Maker Protocol employs a two-token system. The first being, Dai, a collateral backed stablecoin that offers stability. The Maker Foundation and the MakerDAO community believe that a decentralized stablecoin is required to have any business or individual realize the advantages of digital money. Second, there is MKR, a governance token that is used by stakeholders to maintain the system and manage Dai. MKR token holders are the decision-makers of the Maker Protocol, supported by the larger public community and various other external partis.

Maker is a decentralized finance platform that empowers everyone through equal access to the global financial marketplace. The new version of the Maker Protocol, Multi Collateral Dai (MCD), accepts any Ethereum-based asset as collateral for generating Dai, provided it has been approved by MKR holders and given specific risk parameters through the Maker decentralized governance process. This change aims to unlock the power of decentralized finance for everyone.



**Fig 1.3.10: Maker**

Maker ecosystem history began with MakerDAO, founded in 2014 by Rune Christensen, a Danish entrepreneur and graduate of the University of Copenhagen. Christensen co-founded Try China and later transitioned to blockchain after studying international business and biochemistry.

Two venture capital firms, Dragonfly Capital and Paradigm, invested a combined total of $27.5 million in the MakerDAO platform in December 2019, purchasing about 5.5 percent of the total supply of DAI in circulation worldwide.

**Coinbase announces custody service; Maker support planned**

Coinbase Custody announced in March 2019 that it would offer staking services to institutional traders using the proof of stake model. The first coin to use this service is Tezos (XTZ), which stores clients' assets in cold storage to reduce theft risks. However, this service will only work with certain cryptocurrencies, such as bitcoin and Ethereum, which use proof of work governance models. Support for Maker tokens is planned for the future.

**ETH crash causes mass liquidations**

On March 12-13, 2020, the price of Ether fell due to the Maker protocol not updating ETH-USD prices in time, leading to many users winning auctions with 0 DAI bids. This disrupted the protocol, allowing some to exploit price discrepancies and others to lose their Vault stakes. The Maker Foundation held digital token auctions to cover $4.5 million worth of undercapitalized debt created on March 12. Crypto firm Paradigm won 68% of the auctions, gaining over 14,000 MKR tokens. A lawsuit was filed against the Maker Foundation in April, alleging misrepresentation of the protocol's risks. In September, MKR token holders voted 2 to 1 to not compensate Vault holders for losses due to the crash.

**Flash crash loan is used to pass a governance vote**

In late October, a Maker user used a flash crash loan to manipulate a governance vote by borrowing synthetic Ether as collateral to buy $7 million worth of MKR tokens. The tokens were returned after the vote, causing the Maker Foundation to post about the incident on Makerdao's forum, highlighting potential flaws in the platform's governance system.

**1.4 SOURCE**

The Dataset is of Secondary datatype. The Dataset has been collected from the website Kaggle.com: “Cryptocurrencies Daily Prices”. Kaggle is a platform enables users to find and publish datasets. The dataset contains various cryptocurrency prices day-wise.

**1.5 DESCRIPTION OF DATA**

The Dataset contains the daily cryptocurrency prices from 2018-2024(present). The original dataset included 129 different cryptocurrencies each having 6 columns. Some of the cryptocurrencies:

AAVE, Cardano, Basic attention token, Bitcoin Cash, Biconomy, Binance Coin, Bitcoin, Ethereum, Litecoin, Ripple, Maker, Zcash, Pancake Swap, Common Knowledge Base, Curve DAO, Dai, Enjin Coin, Flow, Gnosis Coin, Hedera Hashgraph token, Internet of Things, KuCoin Token, Chainlink, USD Coin, VeChain Token and etc.

The dataset is pre-processed and selected 5 Cryptocurrencies datasets such as Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Zcash (ZEC), Maker (MKR) from the original dataset. Since these are some popular cryptocurrencies.

After pre-processing the dataset each cryptocurrency had the following number of rows and column: Bitcoin dataset had 2739 rows and 6 columns, Ethereum dataset had 2745 rows and 6 columns, Litecoin dataset had 2737 rows and 6 columns, Zcash dataset had 2739 rows and 6 columns, Maker dataset had 2372 rows and 6 columns respectively.

**1.6 FEATURES OF DATA**

The dataset consists of 6 columns (Features) such as Ticker, Date, Open, High, Low, Close.

1. **Ticker**: This column specifies the name of the Cryptocurrency (e.g., BTC, LTC, ETH, ZEC, MKR).
2. **Date**: The date on which the price data was recorded.
3. **Open**: The opening price of the cryptocurrency at the beginning of the day.
4. **High**: The highest price reached by the cryptocurrency during the day.
5. **Low**: The lowest price reached by the cryptocurrency during the day.
6. **Close**: The closing price of the cryptocurrency at the end of the day.

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Ticker 11861 non-null object

1 Date 11861 non-null object

2 Open 11861 non-null float64

3 High 11861 non-null float64

4 Low 11861 non-null float64

5 Close 11861 non-null float64

**1.7 REVIEW OF CHAPTERS**

* Chapter 2 gives the brief introduction about Time Series Analysis and Forecasting techniques using time series analysis like ARIMA, which is used to forecast the future data.
* Chapter 3 helps to understand the data by visualizing the data according to the problem. This also includes the analysis of the data which will give the forecasted data.
* Chapter 4 describes the summary, conclusion and bibliography.

**1.8 PROBLEM STATEMENT**

The project aims to offer investors valuable insights into potential price movements based on historical data and to develop a predictive model that can forecast the future price and finding the best cryptocurrency to invest out of Bitcoin, Ethereum, Litecoin, Zcash, Maker.

**1.8 OBJECTIVES**

* To analyse historical cryptocurrency price data
* To build time series forecasting models such as ARIMA to forecast future cryptocurrency prices.
* To build time series forecasting deep learning neural network model such as LSTM (Long Short-Term Memory).
* To study the current status of cryptocurrency in India and the future it holds.
* To study the changes of cryptocurrency prices and effects for the changes.
* To study which time series forecasting models gives best accuracy out of ARIMA and LSTM.

**Chapter 2**

**METHEDOLOGY**

**2.1 R EVIEW OF TIME SERIES ANALYSIS**

A Time Series is defined as a set of observations on a variable generated sequentially in time. A statistical data of a variable understudy is arranged with reference to time (or) different time periods is called Time Series. The measurement of the variable may be made continuously or may be made at discrete intervals. Often a variable continuous in time is measured at regular intervals and this produces a discrete series of data. Thus, a time series may be represented as a set of data (y1, y2, …, yn), yt denoting the value of the variable y at time t.

The main objectives in investigating a time series are to gain understanding of the process generating the time series and forecasting future values of the observed series. If the objective is to gain understanding of the process, one would be interested to know how the various explanatory variables on which the observed variable is thought to depend are incorporated in the model. If the forecasting of the observed variable is the main objective one uses the simplest model which adequately describes the behaviour of the observed variable and the required forecast is made on the basis of this model. We shall restrict consideration to models in which observations can be expressed as explicit functions of time plus a random error component. More complex models in which current value of the variable can depend additionally on past values itself or past values of related variables are possible.

**2.2 COMPONENTS OF TIME SERIES**

The various factors affecting the values of a variable in a time series are called components of time series. The components od time series can be broadly classified in to the following four categories:

1. Secular trend (or) Trend component
2. Seasonal component
3. Cyclic component
4. Irregular component

**2.2.1 SECULAR TREND (OR) TREND COMPONENT**

This is the long term upward and downward movement of the series due to factors that influence the mean of the series. The general tendency observed in a variable of a time series data of increasing or decreasing during a long time period is called Long-term variations (or) Secular trends. The word “Trend” refers the general moment i.e., either increased (or) decreased of time series over a long period of time. Some series may remain more or less at a constant level. Sudden or frequent changes are incompatible with the idea of the trend. Thus, trend may be defined as a slowly changing non-random component of a time series. In general, this type of variations mostly observed in business and economic statistics.

**2.2.2 SEASONAL COMPONENT**

The variation in a time series is due to some rhythmic forces which occur in a regular (or) periodic manner over a span on less than one year and the same (or) almost same pattern year after year is called Seasonal variations. The seasonal variations occur during the time period less than a year i.e., these may be observed quarterly, monthly, weekly, daily and soon.

This is a measure of the characteristic behaviour of the series during each season in the period, which may be one year. Many time series in business and industry show variations in values from one season to the other, which repeat itself at least approximately, year by year. A year means a time period during which a full cycle of such seasonal variations is completed and seasons are intervals of time constituting this period. Thus, a period may be a span of 12 months and a season a quarter or a month; a period may be a span of 24 hours of a day and a season a three-hourly period, etc. The basic cause of seasonal variation is usually either the seasonal weather, or some fixed period of time during which most activities take place. Seasonality is the non-random component of a time series which tends to repeat itself at regular intervals of time.

**2.2.3 CYCLICAL COMPONENT**

The cyclical fluctuation means the oscillatory movement of a time series, the period of oscillation being more than a year. One complete period is called a cycle. By cyclical component we mean, therefore, the slow oscillatory movement of a time series which repeats itself in each cycle. The cyclic variations are even though more or less regular, but need not be periodic i.e., need not follow exactly the same pattern in a period after period. The cyclic movements occurred in any business activity is known as business cycle, which have four phases i.e., prosperity, recession, depression and recovery, which normally lasts from 7 to 11 years. The length of the cycle and its intensity of fluctuations may vary from one cycle to the other. The length of many cycles average about 3 to 4 years, though some are longer than 15 years. For short-term forecasting, it is desirable to consider the trend and cycle together, as it is impossible to obtain information about cycles without data from several cycles. Any cyclical feature in the data thus gets mingled up with the trend component. Conversely, the trend itself can be regarded as part of a cycle of long duration. In forecasting for a short period trend and cyclical component are taken together. The study of cyclic variations is more useful to formulate suitable politics for the improvement of business activities.

**2.2.4 IRREGULAR COMPONENT**

Some kind of variation which are purely random, unpredictable, accidental and non-recurring again are called Irregular variations. These variations are due to the factors like floods, wars, earthquakes, epidemics, strikes, logouts etc which cannot be predicted and beyond the control of human hand.

**2.3 FORECASTING**

Forecasting is the process of making predictions based on past and present data. Later these can be compared against what happens. Prediction is similar but more general ter. Forecasting might refer to specific formal statistical methods employing time series, cross-sectional or longitudinal data, or alternatively to less formal judgemental methods or the process of prediction and resolution itself. Usage can vary between areas of application: for example, in hydrology the terms “forecast” and “forecasting” are sometimes reserved for estimates of values at certain specific future times, while the term “prediction” is used for more general estimates, such as the number of times floods will occur over a long period.

Risk and uncertainty are central to forecasting and prediction; it is generally considered a good practice to indicate the degree of uncertainty attaching to forecasts. In any case, the data must be up to date in order for the forecast to be as accurate as possible. In some cases, the data used to predict the variable of interest is itself forecast.

Forecasting has application in many situations:

* Economic Forecasting
* Earthquake prediction
* Egain forecasting
* Player and team performance in sports
* Product forecasting
* Sales forecasting
* Political forecasting
* Telecommunications forecasting
* Transport planning and forecasting
* Weather forecasting and soon.

**2.3.1 CATEGORIES OF FORECASTING METHODS**

**Qualitative forecasting:** These techniques are subjective, based on the opinion and judgement of consumers and experts; they are appropriate when past data are not available. They are usually applied to intermediate or long-range decisions. Examples of qualitative forecasting methods are informed opinion and judgement, the Delphi method, market research and historical life-cycle analogy.

**Quantitative forecasting:** These models are used to forecast future data as a function of past data. They are appropriate to use when past numerical data is available and when it is reasonable to assume that some of the patterns in the data are expected to continue into the future. These methods are usually applied to short or intermediate-range decisions. Examples of quantitative forecasting methods are last period demand, simple and weighted N-period moving averages, simple exponential smoothing, Poisson process model forecasting and multiplicative seasonal indexes. Previous research shows that different methods may lead to different level of forecasting accuracy. For example, GMDH neural network was found to have better forecasting performance than the classical forecasting algorithms such as Single Exponential Smooth, Double Exponential Smooth, ARIMA and back-propagation neural network.

**2.3.2 FORECASTING TIME SERIES**

The use at time *t* of available observations from a time series to forecast its value at some future time *t + l* can provide a basis for

1. Economic and business planning
2. Production planning
3. Inventory and production control
4. Control and optimization of industrial processes

As originally described by Holt et al. (1963), Brown (1962), and the Imperial Chemical Industries (ICI) monograph on short term forecasting (Couties, 1964), forecasts are usually needed over a period known as the *lead time*, which varies with each problem. For example, the lead time in the inventory control problem was defined by Harrison (1965) as a period that begins when an order to replenish stock is placed with the factory and lasts until the order is delivered into stock.

We will assume that observations are available at *discrete*, equispaced intervals of time. For example, in a sales forecasting problem, the sales Zt in the current month *t* and the sales Zt-1, Zt-2, Zt-3, … in previous months might be used to forecast sales for lead times *l* = 1, 2, 3, …,12 months ahead. Denote by Zˆt (*l*) the forecast made at *origin t* of the sales Zt+1 at some future time *t + l*, that is, at *lead time l*. The function Zˆt (*l*), which provides the forecasts at origin t for all future lead times, based on the available information from the current and previous values Zt, Zt-1, Zt-2, Zt-3, …through time t, will be called the *forecast function* at origin *t*. Our objective is to obtain a forecast function such that the mean square of the deviations Zt+1 - Zˆt (*l*) between the actual and forecasted values is as small as possible for each *lead time l*.

In addition to calculating the best forecasts, it is also necessary to specify their accuracy, so that, for example, the risks associated with decisions based upon the forecasts may be calculated. The accuracy of the forecasts may be expressed by calculating *probability limits* on either side of each forecast. These limits may be calculated for any convenient set of probabilities, for example, 50 and 95%. They are such that the realized value of the time series, when it eventually occurs, will be included within these limits with the stated probability.

**2.3.3 FORECASTING ACCURACY**

The forecast error (residual) is the difference between the actual value and the forecast value for the corresponding period:

*Et = Yt - Ft*

Where E is the forecast error at period t, Y is the actual value at period t, and F is the forecast for period t.

A good forecasting method will yield residuals that are uncorrelated. If there are correlations between residual values, then there is information left in the residuals which should be used in computing forecasts. This can be accomplished by computing the expected value of a residual as a function of the known past residuals, and adjusting the forecast by the amount by which this expected value differs from zero.

A good forecasting method will also have zero mean. If the residuals have a mean other than zero, then the forecasts are biased and can be improved by adjusting the forecasting technique by an additive constant that equals the mean of the unadjusted residuals.

**2.4 STATIONARY TIME SERIES**

A Stationary process is a stochastic process whose unconditional joint probability distribution does not change when shifted in time. Consequently, parameters such as mean and variance also do not change over time.

Since stationarity is an assumption underlying many statistical procedures used in time series analysis, non-stationary data are often transformed to become stationary. The most common cause of violation of stationarity is a trend in the mean, which can be due either to the presence of a unit root or of a deterministic trend. In the former case of a unit root, stochastic shocks have permanent effects, and the process is not mean reverting. In the latter case of a deterministic trend, the process is called a trend-stationary process, and stochastic stocks have only transitory effects after which the variable tends toward a deterministically evolving mean.

A trend stationary process is not strictly stationary, but can easily be transformed into a stationary process by removing the underlying trend, which is solely a function of time. Similarly, processes with one-or-more unit roots can be made stationary through differencing. An important type of non-stationary process that does not include a trend-like behaviour is a cyclostationary process, which is a stochastic process that varies cyclically with time.

**2.4.1 STRONG STATIONARITY**

Strong stationarity demands that the joint probability distribution of any finite set of observations in the time series remains unchanged, no matter how the timeline shifts. It requires the shift-invariance of the finite-dimensional distributions of a stochastic process. This means that the distribution of a finite sub-sequence of random variables of the stochastic process remains the same as we shift it along the time index axis.

**2.4.2 WEAK STATIONARITY**

Weak stationarity refers to a time series where the mean, variance, and autocorrelation are constant over time, but higher-order moments may vary. It only requires the shift-invariance of the first moment and the cross moment. This means the process has the same mean at all time points, t and t-k, depend only on k, the difference between the two times, and not on the location of the points along the time axis.

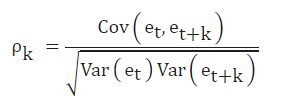
**2.5 AUTO-CORRELATION FUNCTION & PARTIAL AUTO-CORRELATION FUNCTION**

**Auto-correlation function (ACF):**

Autocorrelation, sometimes known as serial correlation in the discrete time case, is the correlation of a signal with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations of a random variable as a function of the time lag between them. The analysis of autocorrelation is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal obscured by noise, or identifying the missing fundamental frequency in a signal implies by its harmonic frequencies. It is often used in signal processing for analysing functions or series of values, such as time domain signals.

Unit root processes, trend-stationary processes autoregressive processes, and moving average processed are specific forms of processes with autocorrelation.

The auto correlation function (ACF) reveals how the correlation between any two values of the signal changes as their separation changes. It is a time domain measure of the stochastic process memory, and does not reveal any information about the frequency content of the process. Generally, for an error signal, et, the ACF is defined as,



A correlated process, such that ARMA or ARIMA, has non-zero values at lags other than zero to indicate a correlation between different lagged observations.

ACF plots the correlation coefficient against the lag, and it’s a visual representation of autocorrelation.

The correlation coefficient is measured either by Pearson’s correlation coefficient or by Spearman’s rank correlation coefficient. The correlation coefficient can range from -1 to +1.

**Partial Auto-correlation function (PACF):**

In time series analysis, the partial autocorrelation function (PACF) gives the partial correlation of a stationary time series with its own lagged values, regressed the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control for other lags. The PACF is a statistical tool that measures how well a time series is correlated with its past values, while ignoring the effects of values in between.

This function plays an important role in data analysis aimed at identifying the extent of the lag in an autoregressive model. The use of this function was introduced as part of the Box-Jenkins approach to time series modelling, whereby plotting the partial autocorrelative functions one could determine the appropriate lags p in an AR (p) model or in an extended ARIMA (p, d, q) model.

The PACF graph is constructed by plotting all the values of PACF obtained from regressions at different lags.

**2.5.1 IMPORTANCE OF ACF & PACF**

ACF and PACF graphs are used to find out the order of AR and MA component of an ARIMA model.

If the ACF graph is declining and there are a few significant lags in the PACF, then this indicates the process is Auto-regressive (AR). We can select order p for AR (p) model based on significant spikes from the PACF plot. Spikes those are outside the blue boundary of the PACF plot tell us the order of the AR model.

If the PACF graph is declining and there are a few significant lags in the ACF, then this indicates the process is Moving average (MA). We can select the order q for MA (q) model based on significant spikes from the ACF plot. Spikes those are outside the blue boundary of the ACF plot tell us the order of the MA model.

The blue area in the ACF and PACF graphs indicated 95% confidence interval and it is an indicator of significance threshold. Anything within the area is statistically close to zero and anything outside is statistically non-zero.

To determine the order of the model, we have to consider the spikes which are outside the significance threshold (blue area).

**2.6 AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODEL (ARIMA)**

An Autoregressive Integrated Moving Average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the dataset or to predict future trends. Autoregressive Integrated Moving Average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. To better comprehend the data or to forecast upcoming series points, both of these models are fitted to time series data. ARIMA models are applied in some cases where data show evidence of non-stationarity in the sense of expected value, where an initial differencing step can be applied one or more times to eliminate the non-stationarity of the mean function.

Non-seasonal ARIMA models are generally denoted ARIMA (p, d, q) where parameters p, d, q are non-negative integers, p is the order of the autoregressive model, d is the degree of differencing and q is the order of the moving average model.

An autoregressive integrated moving average model is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The model’s goal is to predict future securities or financial market moves by examining the differences between values in the series instead of through actual values.

An ARIMA model can be understood by outlining each of its components as follows:

* **Autoregression (AR)**: refers to a model that shows a changing variable of interest that regresses on its own lagged, or prior values.
* **Integrated (I)**: represents the differencing of raw observation to allow the time series to become stationary the data values have been replaced with the difference between their values and previous values.
* **Moving average (MA)**: incorporated the dependency between an observation and a residual error from a moving average model applied to lagged observation. The regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.

**2.6.1 STEPS FOR ARIMA IMPLEMENTATION**

The general steps to implement an ARIMA model are –

1. **Load the data:** The first step for model building is of course to load the dataset.
2. **Check Stationarity:** In order to satisfy the assumption, it is necessary to make the series stationary. This would include checking the stationarity of the series and performing required transformations.

There are three basic criteria for a series to be classified as stationary series. First, the mean of the series should not be a function of time, this means that the mean should be roughly constant through some variance can be modelled. Second, the variance of the series should not be a function of time. This property is known as the homoscedasticity. Third, the covariance of the terms should not be a function of time.

A stationary time series is a flat looking series, without trend, constant variance over time, a constant autocorrelation structure over time, and no periodic fluctuations.

We can confirm that our time series is non-stationary by using two tests: the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. The ADF test is a type of statistical test called unit root test. The ADF test is conducted with the following assumptions:

* The null hypothesis for the test is that the series is non-stationary, or series has a unit root.
* The alternative hypothesis for the test if that the series is stationary, or series has no unit root.

If the null hypothesis is not rejected, then the test may provide evidence that the series is non-stationary. This means that when you run the ADF test, if the rest statistic is less than the critical value of p-value is less than 0.05, then we reject the null hypothesis, which means the time series is stationary. If the null hypothesis is accepted then we have to difference data and convert non-stationary data into stationary data.

1. **Create Autocorrelation and partial autocorrelation plots:** ACF and PACF plots are used to determine the input parameters for our ARIMA model. Autocorrelation tells whether the data exhibit randomness and how related one observation is to an immediately adjacent observation. This gives us a sense of what sort of model might best represent the data. The autocorrelations are often plotted to see the correlation between the point up to and including the lag unit. There are two approaches to autocorrelation: the autocorrelation function (ACF) and the partial autocorrelation function (PACF).

In ACF, the correlation coefficient is in the x-axis whereas the number of lags is shown in the y-axis.

The PACF is helpful in determining the order of the AR part of the ARIMA model. It is also useful to determine or validate how many seasonal lags to include in the forecasting equation of a moving average based forecast model for a seasonal time series. This is called the seasonal moving average.

1. **Determine the p, d, q values:** Read the values of p and q from the plots in the previous step. Identify the value of p from the PACF plot and value of q from the ACF plot. The value of d is based on the number of times the data is differenced to convert in to stationary.
2. **Fit ARIMA model:** Using the processed data and parameter values we calculated from the previous steps, fit the ARIMA model and predict the future values.
3. **Calculate RMSE:** To check the performance of the model, check the Root Mean Square Error (RMSE) value using the predictions and actual values on the validation set.

**2.6.2 AUTO-REGRESSIVE MODEL**

In statistics, economics, and signal processing, and Auto-regressive (AR) model is a representation of a type of random process; as such it can be used to describe certain time-varying processes in nature, economics, behaviour, etch. The autoregressive model specifies that the output variable depends linearly on its own previous values and on a stochastic form; thus, the model is in the form of a stochastic difference equation which should not be confused with a differential equation.

Autoregression is a statistical technique used in time-series analysis that assumes that the current value of a time series is a function of its past values. Autoregressive models use similar mathematical techniques to determine the probabilistic correlation between elements in a sequence.

Auto-regressive models are multiple regression models applied on lag series generated using the original time series. In multiple linear regression, the output is a linear combination of multiple input variables. In case of Autoregressive models, the output is the future data point and it can be expressed as a linear combination for past p data points. p is the lag window. Thus, an autoregressive model of order p can be written as



Here,

at is the random error or white noise.

ɸ1, ɸ2, …, ɸp are the finite set of weighted parameters.

This process is called an Auto-regressive process of order p or AR (p) process.

The AR (p) model can be written in the equivalent form



(Or)



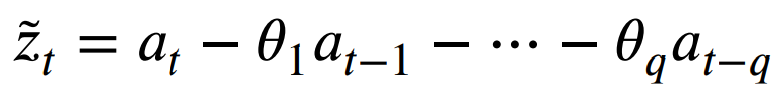
Where ɸ (B) is the Auto-regressive operator

Auto-regressive models are remarkably flexible at handling a wide range of different time series patterns.

**2.6.3 MOVING AVERAGE MODEL**

In time series analysis, the moving average model also known as moving average process, is a common approach fir modelling univariate time series.

Moving Average Models are a type of time series analysis model usually used in econometrics to forecast trends and understand patterns in time series data. In moving average models, the present value of the time series depends on the linear combination of the past white noise error terms of the time series. In time series analysis moving average is denoted by the letter q which represents the order of the moving average model, or in simple words we can say the current value of the time series will depend on the past q error terms.

****

Here,

at, at-1, at-q are the random error or white noise terms associated with the time series at time t, t-1, …, t-q.

 are the moving average constants.

This process is called Moving average process of order q or MA (q) process.

**2.6.4 DIFFERENCING**

Differencing in statistics is a transformation applied to a non-stationary time-series in order to make it stationary in the mean sense, but having nothing to do with the non-stationarity of the variance or autocovariance. Likewise, the seasonal differencing is applied to a seasonal time series to remove the seasonal component. From the perspective of signal processing, especially the Fourier spectral analysis theory, the trend is the low-frequency part in the spectrum of a non-stationary time series, while the season is the periodic frequency part in the spectrum of it. Therefore, the differencing works as a high-pass filter and the seasonal-differencing as a comb filter to suppress the low-frequency trend and the periodic-frequency season in the spectrum domain, respectively.

Differencing removes the changes in the level of a time series, eliminating trend and seasonality and consequently stabilizing the mean of the time series. Sometimes, it may be necessary to difference the data a second time to obtain a stationary time series, which is referred to as second-order differencing.

**2.6.5 AUTO ARIMA**

Auto ARIMA is a statistical algorithm used for time series forecasting. It automatically determines the optimal parameters for an ARIMA model, such as the order of differencing autoregressive (AR) terms, and moving average (MA) terms. Auto ARIMA searches through different combinations of these parameters to find the best fit for the given time series data. This automated process saves time and effort, making it easier for users to generate accurate forecasts without requiring extensive knowledge of time series analysis.

**2.7 HISTORY OF LONG SHORT-TERM MEMORY (LSTM)**

**1989:** Mile Mozer’s work on “focused back-propagation” anticipates aspects of LSTM, which the LSTM paper cites.

**1990:** Recurrent Neural Networks (RNN) were widely used for modelling sequential data, but they suffered from the vanishing gradient problem, which made it difficult to learn long-term dependencies.

**1991:** Sepp Hochreiter analysed the vanishing gradient problem and developed principles of the method in his German diploma thesis, which was considered highly significant by his supervisor Jurgen Schmidhuber.

**1995:** “Long Short-Term Memory (LSTM)” is published in a technical report by Sepp Hochreiter and Jurgen Schmidhuber.

**1996:** LSTM is published at NIPS’1996, a peer-reviewed conference.

**1997:** The main LSTM paper is published in the journal Neural Computation. Hochreiter and Schmidhuber introduced LSTM as a solution to the vanishing gradient problem. They proposed a novel architecture with memory cells and gates to control the flow of information. The initial version of LSTM block included cells, input and output gates.

**1999:** Felix Gers, Jurgen Schmidhuber, and Fred Cummins introduced the forget gate (also called as keep gate) into the LSTM architecture, enabling the LSTM to reset its own state.

**2000:** Gers, Schmidhuber, and Cummins added peephole connections into the architecture. Additionally, the output activation function was omitted.

**2001:** Gers and Schmidhuber trained LSTM to learn languages unlearnable by traditional models such as Hidden Markov Models.

**2004:** First successful application of LSTM to speech recognition.

**2005:** First publication of LSTM with full backpropagation through time and bi-directional LSTM. Daan Wierstra, Faustino Gomez, and Schmidhuber trained LSTM by neuroevolution without a teacher.

**2006:** Graves, Fernandez, Gomez, and Schmidhuber introduce a new error function for LSTM: Connectionist Temporal Classification (CTC) for simultaneous alignment and recognition of sequences. CTC-trained LSTM led to breakthroughs in speech recognition. Felix Gers and Jurgen Schmidhuber published a paper on “Learning Precise Timing with LSTM Recurrent Neural Networks”, which further explored the capabilities of LSTM.

**2007:** Wierstra, Foerster, Peters, and Schmidhuber trained LSTM by policy gradients for reinforcement learning without a teacher.

**2009:** An LSTM trained by CTC won the ICDAR connected handwriting recognition competition. Three such models were submitted by a team led by Alex Graves. One was the most accurate model in the competition and another was the fastest. This was the first time an RNN won international competitions. Justin Bayer, introduced neural architecture search for LSTM.

**2010:** The rise of deep learning led to a resurgence of interest in LSTM, and it became a widely used architecture for modelling sequential data in various domains, including NLP, speech recognition, and time series analysis.

**2013:** Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton used LSTM networks as a major component of a network that achieved a record 17.7% phoneme error rate on the classic TIMIT natural speech dataset.

**2014:** Kyunghyun Cho, put forward a simplified variant of the forget gate LSTM called Gated recurrent unit (GRU). The paper “Deep Learning with COTS HPC Systems” demonstrated the effectiveness of the LSTM on large-scale speech recognition tasks.

**2015:** Google started using an LSTM trained by CTC for speech recognition on Google voice. According to the official blog post, the new model cut transcription errors by 49%. Rupesh Kumar Srivastava, Klaus Greff, and Schmidhuber used LSTM principles to create the Highway network, a feedforward neural network with hundreds of layers, much deeper than previous networks. 7 months later, Kaiming He, Xiangyu Zhang; Shaoqing Ren, and Jian Sun won the ImageNet 2015 competition with an open-gated or gateless Highway network variant called Residual neural network. This has become the most cited neural network of the 21st century. The paper “Long Short-Term Memory Networks for Machine Reading” applied LSTM to machine reading tasks, achieving state-of-the-art results. Since then, LSTM has become a fundamental component of many state-of-the-art models in various fields, including language modelling, text generation, and sequence-to-sequence tasks.

**2016:** Google started using an LSTM to suggest messages in the Allo conversation app. In the same year, Google released the Google Neural Machine Translation system for Google Translate which used LSTMs to reduce translation errors by 60%. Apple announced in its Worldwide Developers Conference that it would start using the LSTM for quick type in the iPhone and for Sin. Amazon released Polly, which generates the voices behind Alexa, using a bidirectional LSTM for text-to-speech technology.

**2017:** Facebook performed some 4.5 billion automatic translations every day using long short-tern memory networks. Researchers from Michigan State University, IBM Research, and Cornell University published a study in the Knowledge Discovery and Data Mining (KDD) conference. Their Time-Aware LSTM performs better on certain data sets than standard LSTM. Microsoft reported reaching 94.9% recognition accuracy on the Switchboard corpus, incorporating a vocabulary of 165,000 words. The approach used “dialog session-based long-short-term memory.

**2018:** OpenAI used LSTM trained by policy gradients to beat humans in the complex video game of Dota 2, and to control a human-like robot hand that manipulated physical objects with unprecedented dexterity.

**2019:** DeepMind used LSTM trained by policy gradients to excel at the complex video game of StarCraft II.

**2021:** According to Google Scholar in 2021, LSTM was cited over 16,000 times within a single year. This reflects applications of LSTM in many different fields including healthcare.

**2024:** An evolution of LSTM called xLSTM is published by a team lead by Sepp Hochreiter.

**2.8 LONG SHORT-TERM MEMORY (LSTM)**

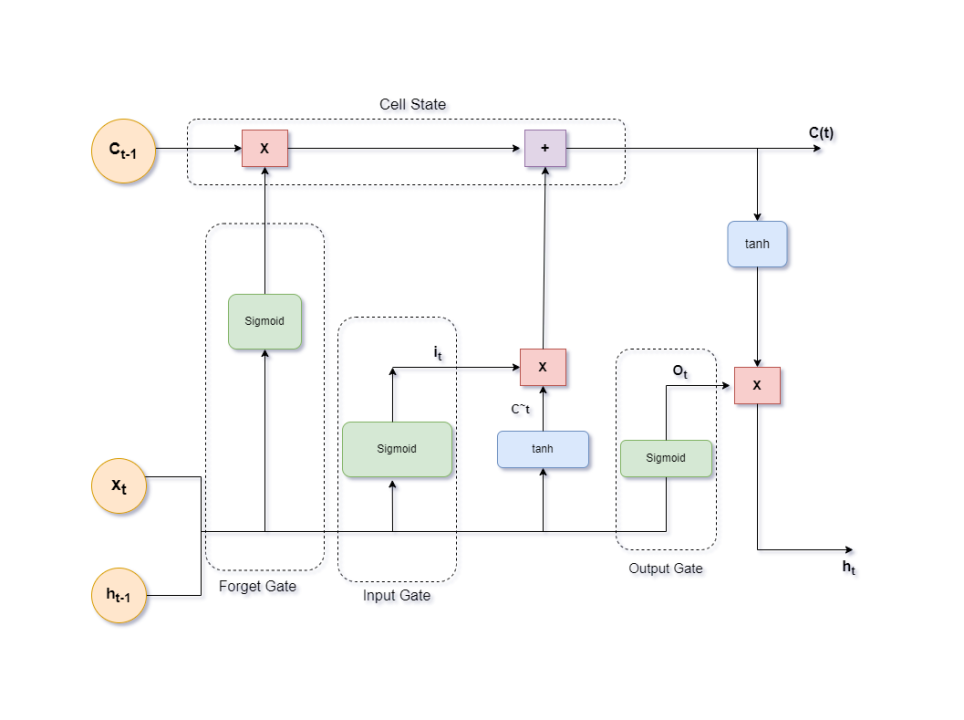
LSTMs are Long Short-Term Memory networks that use Artificial neural networks (ANN) in the field of Artificial Intelligence (AI) and deep learning. Long Short-Term memory (LSTM) is a type of recurrent neural network (RNN) aimed at dealing with the vanishing gradient problem present in traditional RNNs. It is an improved version of recurrent neural network designed by Hochreiter & Schmidhuber.

Long Short-Term Memory (LSTM) is a special type of neural network designed to overcome the limitations of traditional neural networks. It is particularly useful for tasks that involve sequential data, such as speech, text, or time series data. The main advantage of LSTM is their ability to remember information for a long time, making them ideal for tasks that require a “short-term memory” that can last thousands of time steps. The name is made in analogy with long-term memory and shirt-term memory and their relationship, studied by cognitive psychologists since early 20th century. They use a memory cell and gates to control the flow of information, allowing them to selectively retain or discard information as needed and thus avoid the vanishing gradient problem that plagues traditional RNNs.

It is applicable to classification, processing and predicting data based on time series such as in handwriting, speech recognition, machine translation, speech activity detection, robot control, video games and healthcare.

**2.8.1 LSTM ARCHITECTURE**

The LSTM architectures involve the memory cell which is controlled by three gates: the input gate, the forget gate, and the output gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies.

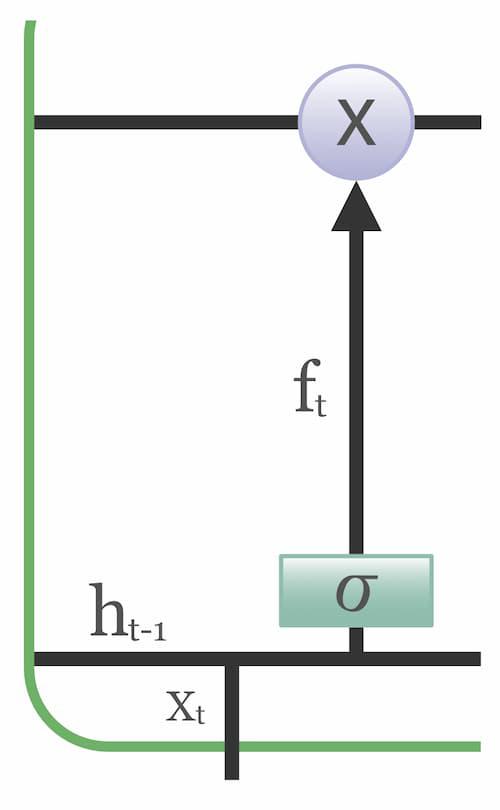


**Fig 2.1: LSTM Architecture**

Networks in LSTM architectures can be stacked to create deep architectures, enabling the learning of even more complex patterns and hierarchies in sequential data. Each LSTM layer in a stacked configuration captures different levels of abstraction and temporal dependencies within the input data.

**Forget Gate:**

The forget gate controls what information is removed from the memory cell.



**Fig 2.2: Forget gate**

The information that is no longer useful in the cell state is removed with the forget gate. It decides what information to discard from the previous state by mapping the previous state and the current input to a value between 0 and 1.

Two inputs xt (input at the particular time)and ht-1 (previous cell output) are fed to the gate and multiplies with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

The equation for the forget gate is:

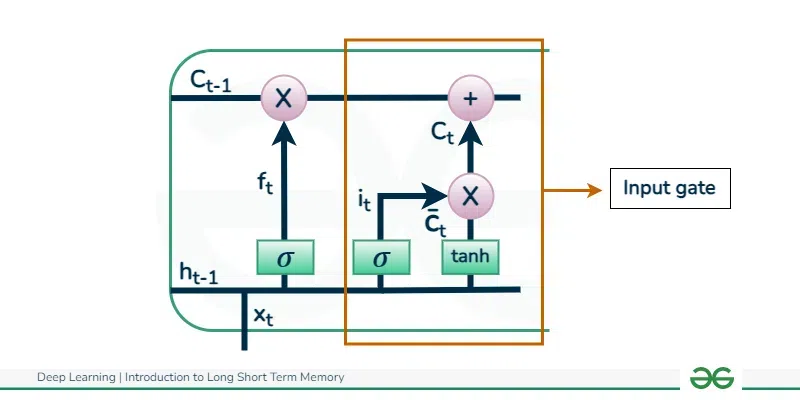


Where:

* Wf represents the weight matrix associated with the forget gate.
* [ht-1, xt] denotes the concatenation of the current input and the previous hidden state.
* Bf is the bias with the forget gate
* σ is the sigmoid activation function

**Input Gate:**

The input gate controls what information is added to the memory cell.



**Fig 2.3: Input Gate**

The addition of useful information to the cell state is done by the input gate. It decides which pieces of new information to store in the current cell state, using the same system as forget gates.

First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs ht-1 and xt. Then, a vector is created using tanh function that gives an output from -1 to +1, which contains all the possible values from ht-1 and xt. At last, the values of the vector and the regulated values are multiplied to obtain the useful information.

The equation for the input gate is:



****

We multiply the previous state by ft, disregarding the information we had previously chosen to ignore. Next, we include it∗Ct. This represents the updated candidate values, adjusted for the amount that we chose to update each state value.



Where:

* ⊙ denotes element-wise multiplication
* tanh is tanh activation function

**Output Gate:**

The output gate controls what information is output from the memory cell.



**Fig 2.4: Output Gate**

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. It controls which piece od information in the current cell state to output by assigning a value from 0 to 1 t the information, considering the previous and current states.

First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs ht-1and xt. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

The equation for the output gate is:



**2.8.2 ADVANTAGES OF LSTM**

* **Handling Long Sequences:** LSTMs are well-suited for processing sequences of data with long-range dependencies. They can capture information from earlier time steps and remember it for a more extended period, making them effective for tasks like natural language processing (NLP) and time series analysis.
* **Avoiding Vanishing Gradient Problem:** LSTMs address the vanishing gradient problem, which is a common issue in training deep networks, particularly RNNs. The architecture of LSTMs includes gating mechanisms (such as the forget gate) that allow them to control the flow of information and gradients through the network, preventing the gradients from becoming too small during training.
* **Handling Variable-Length Sequences:** LSTMs can handle variable-length input sequences by dynamically adjusting their internal state. This is useful in many real-world applications where the length of the input data varies.
* **Memory Cell:** LSTMs have a memory cell that can store and retrieve information over long sequences. This memory cell allows LSTMs to maintain important information while discarding irrelevant information, making them suitable for tasks that involve remembering past context.
* **Gradient Flow Control:** LSTMs are equipped with mechanisms that allow them to control the flow of gradients during backpropagation. The forget gate, for example, can prevent gradients from vanishing when they need to be propagated back in time. This enables LSTMs to capture information from earlier time steps effectively.

**2.9 DATA VISUALISATION**

Data visualization is a crucial aspect of machine learning that enables analysts to understand and make sense of data patterns, relationships, and trends. Through data visualization, insights and patterns in data can be easily interpreted and communicated to a wider audience, making it a critical component of machine learning. In this article, we will discuss the significance of data visualization in machine learning, its various types, and how it is used in the field.

**Types of Data Visualization:**

The types of data visualization we have used here are Bar chart, Line graph.

**Bar Charts:**

Bar Charts Bar charts are a simple but highly effective way of plotting categorical data against discrete values. The heights (or widths) of the bars are in direct proportion to the values they represent. This makes bar charts an excellent way of comparing discrete variables at a glance. Some bar charts cluster bars into groups of two or three (or more) allowing you to compare numerous variables at different points in time. Another variation is the stacked bar chart, which divides each bar into separate sub-bars, one stacked on top of another. This allows for the introduction of additional variables.

**Line Graphs:**

A line graph-also known as a line plot or a line chart-is a graph that uses lines to connect individual data points. A line graph displays quantitative values over a specified time interval. It displays the data that changes continuously concerning time. In a line graph data points are connected with an edge and data points are represented either with points. It displays information as a series of data points called ‘markers’ connected by straight line segments. It is a basic type of chart common in many fields. It is similar to a scatter plot expect that the measurement points are ordered and joined with straight like segments. A line chart is often used to visualize a trend in data over intervals of time-a time series0 thus the line is often drawn chronologically. In these cases they are known as run charts.

**Chapter 3**

**ANALYSIS OF DATA AND RESULTS**

**3.1 LIBRARIES USED**

**1. NumPy:**

NumPy isa library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It also has functions for working in domain of linear algebra, Fourier transformation, and matrices. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. In 2005, Travis Oliphant created NumPy by incorporating features of competing NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

**2. Pandas:**

Pandas’ library is used for data manipulation and analysis. Pandas is a powerful and versatile library that simplifies the tasks of data manipulation in Python. Pandas is well-suited for working with tabular data, such as spreadsheets or SQL tables. It is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**3. Matplotlib:**

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib’s primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyse and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers. It is Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. It can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

**4.Seaborn:**

Seaborn is a visualization library for statistical graphics plotting in Python. Seaborn is a well-known Python library for data visualization that offers a user-friendly interface for producing visually appealing and informative statistical graphics. It is designed to work with Pandas data frames, making it easy to visualize and explore data quickly and effectively. Seaborn offers a variety of powerful tools for visualizing data, including scatter plots, line plots, bar plots, heat maps, and many more. it also provides support for advanced statistical analysis, such as regression analysis, distribution plots, and categorical plots.

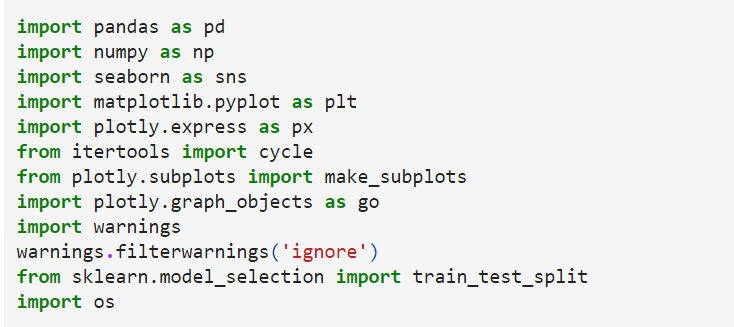
**5.Tensorflow:**

TensorFlow is free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications. It was developed by the Google Brain team for Google’s internal use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released an updated version, TensorFlow 2.0, in September 2019.

**3.3 PYTHON CODE**

**LONG SHORT-TERM MEMORY:**

**Importing Libraries:**

****

**Importing dataset:**

**Bitcoin**

**Ethereum**

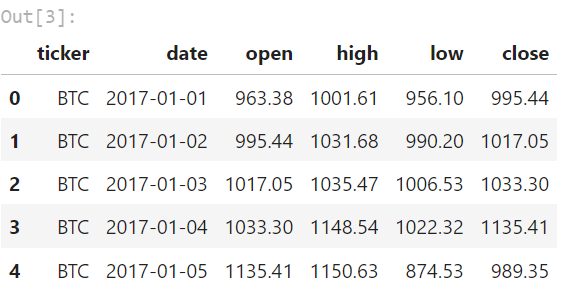
**Litecoin**

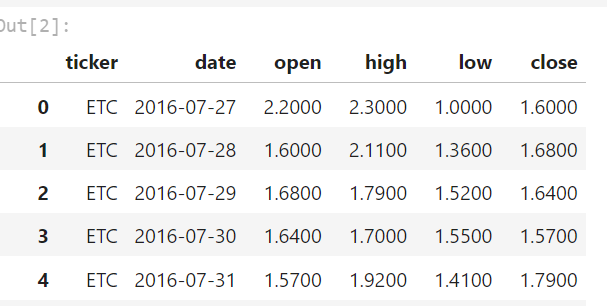
**Maker**

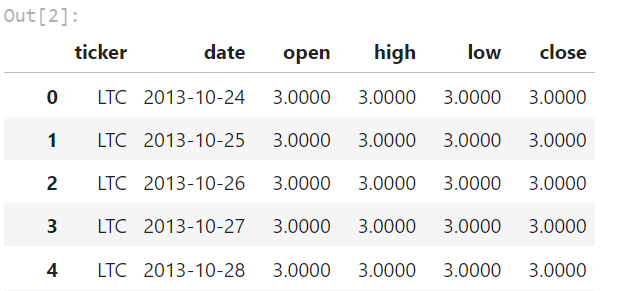
****

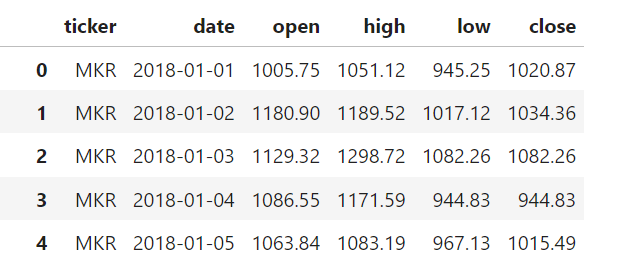
**Reading Head of the data:**

****

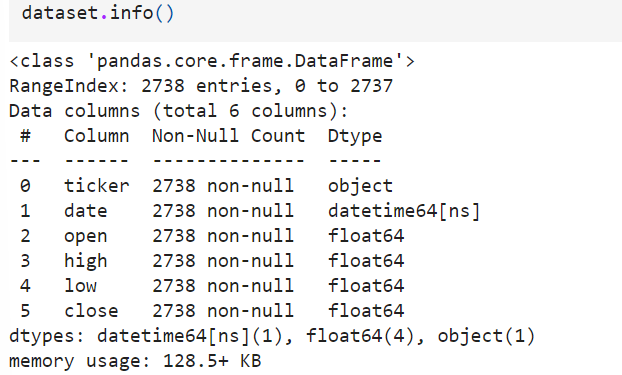
****

****

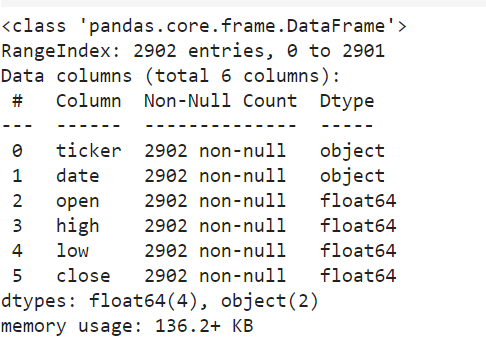
****

****

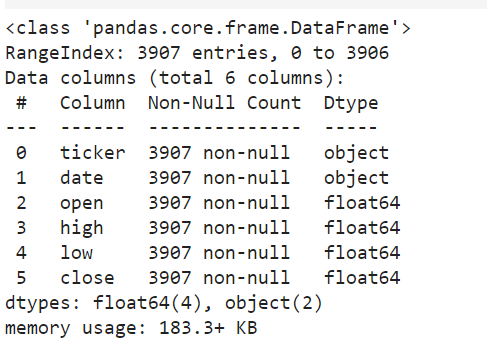
**Information of the dataset:**

****

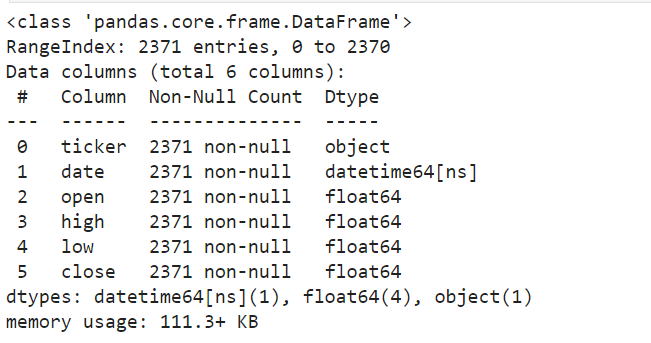
**Ethereum**

****

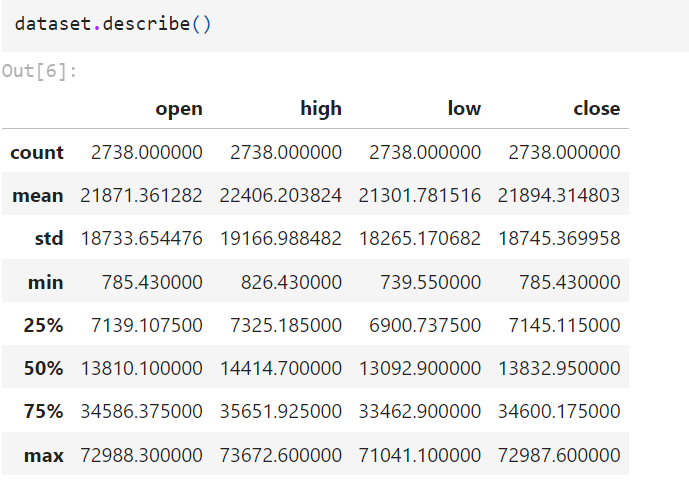
**Litecoin**

****

**Maker**

****

**Describing the data:**

****

**Ethereum**

****

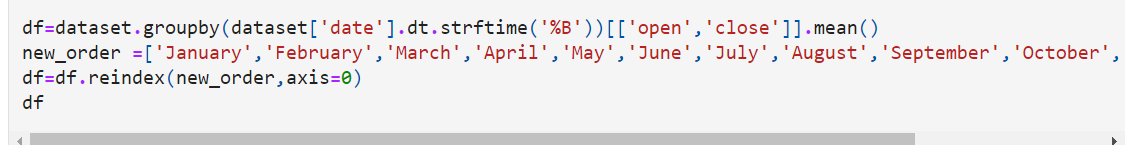
**Litecoin**

****

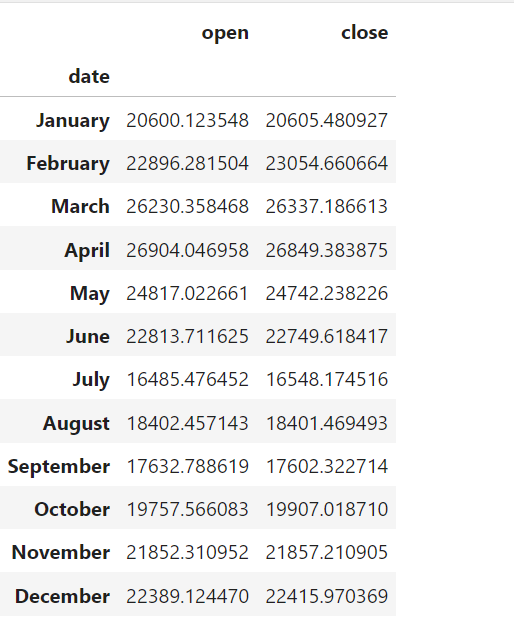
**Maker**

****

**Month wise comparison between open and close prices:**

****

**Bitcoin**

****

**Ethereum**

****

**Litecoin**

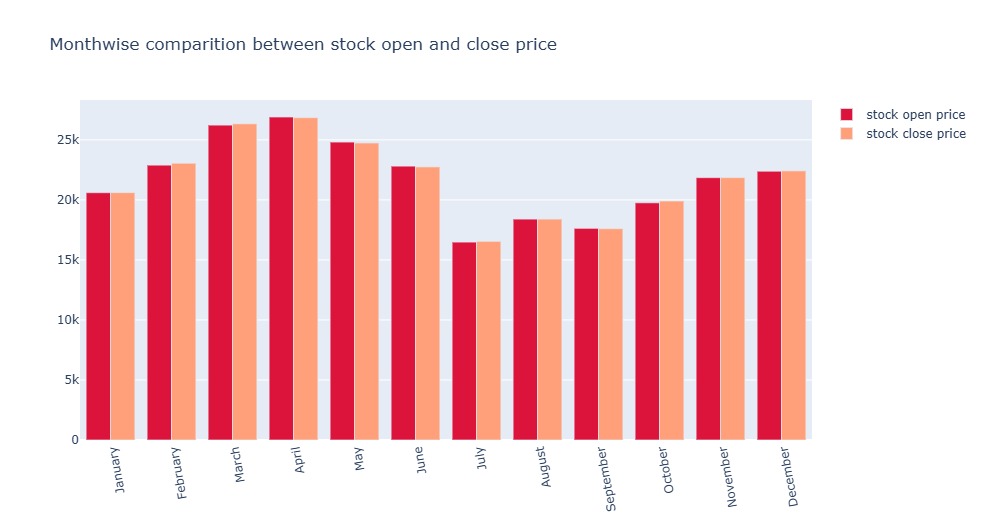
****

**Maker**

****

****

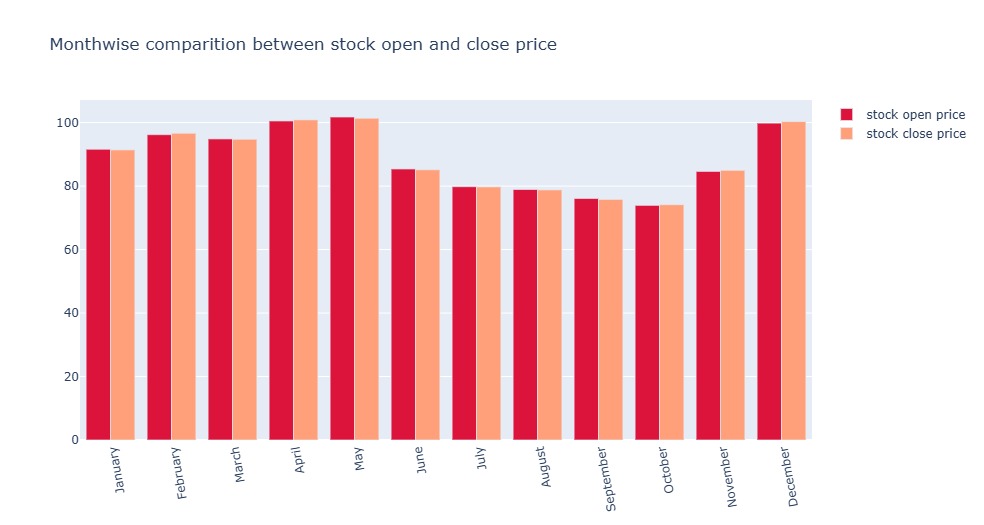
**Bitcoin**



**Ethereum**

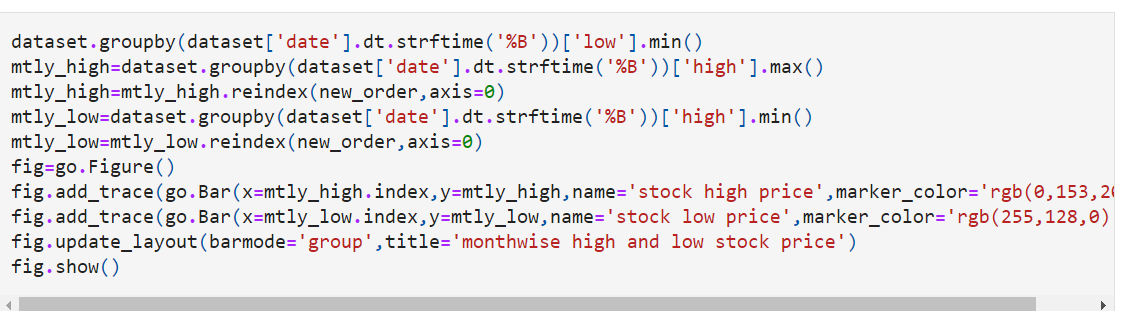


**Litecoin**



**Maker**

**Month wise maximum and minimum high prices:**

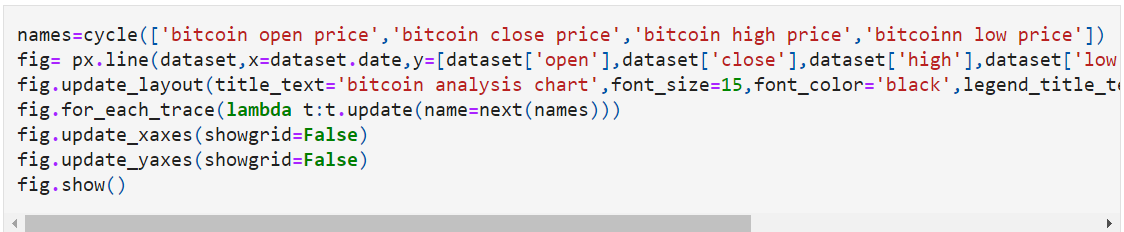
****

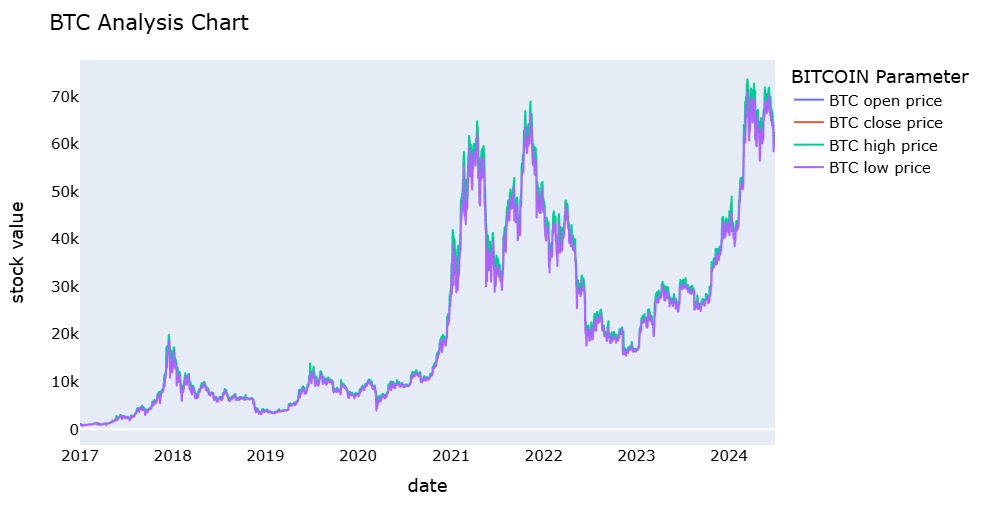
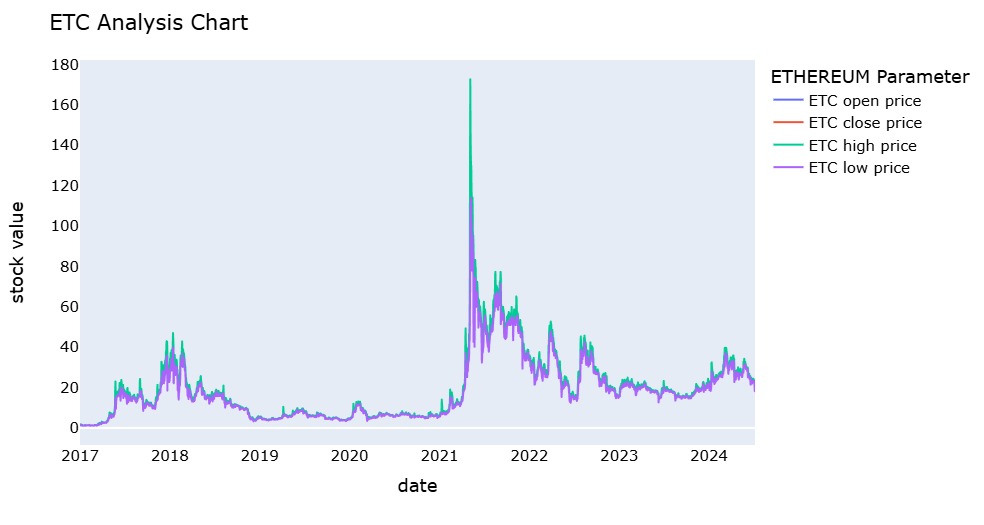
**Bitcoin****Ethereum**

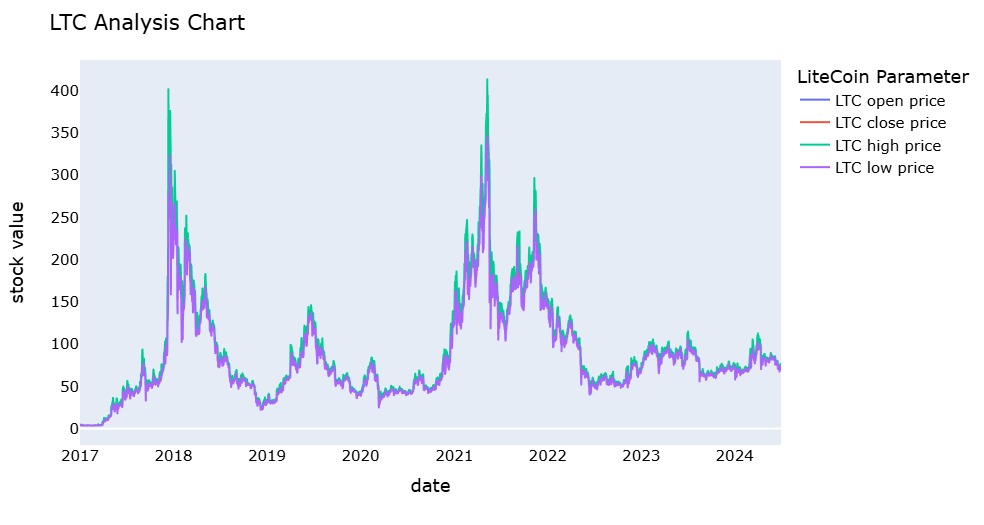
**Litecoin**

**Maker**

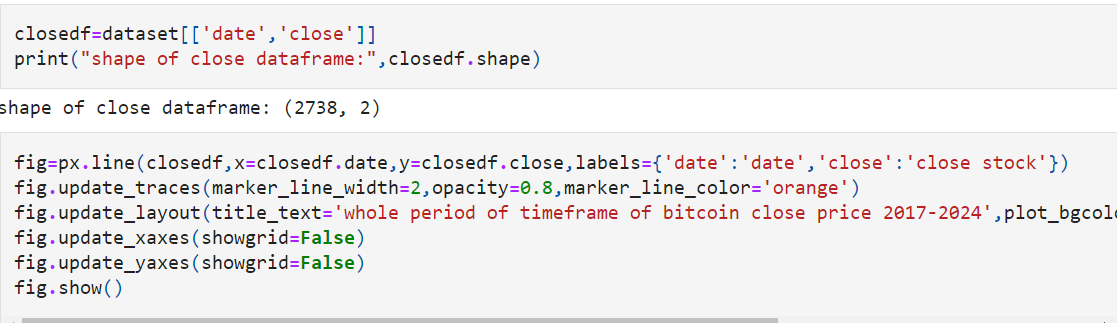
**Analysis Chart:**

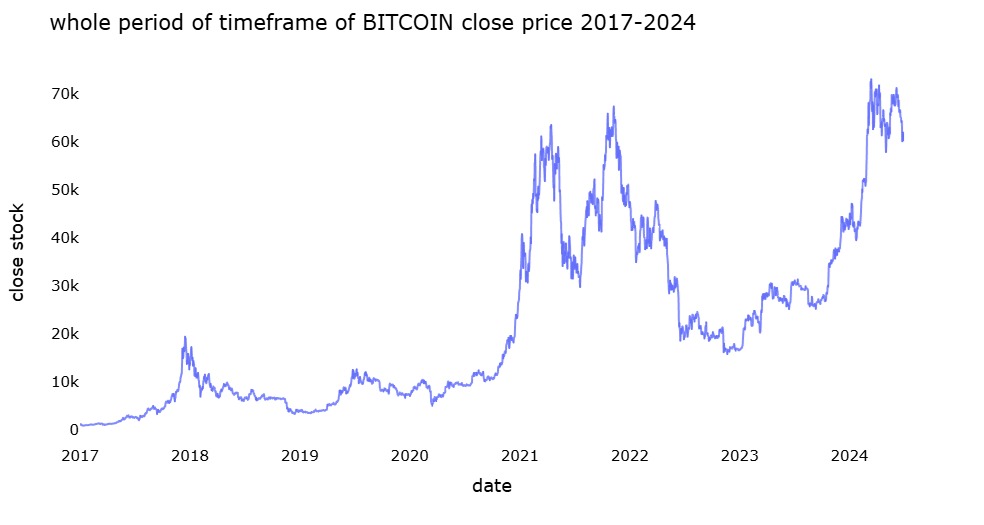
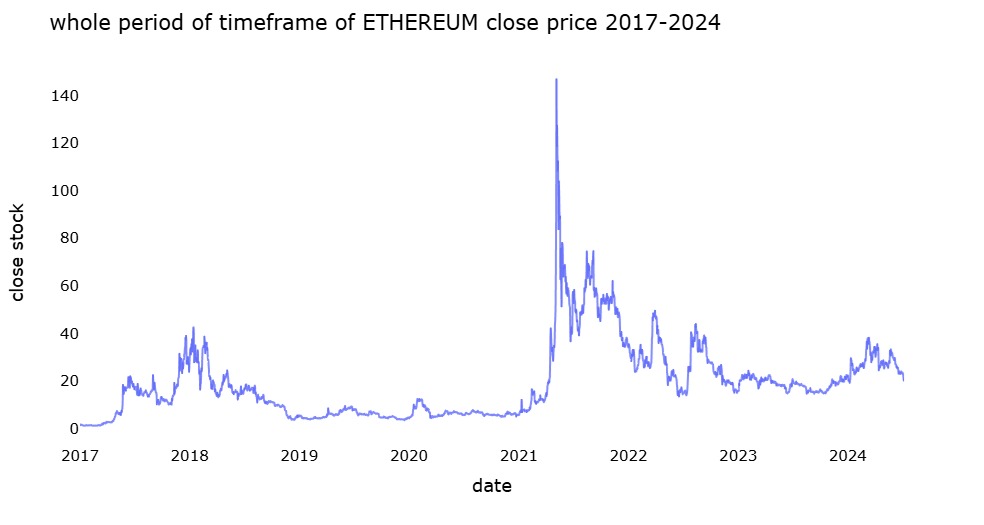
****

**Bitcoin****Ethereum**

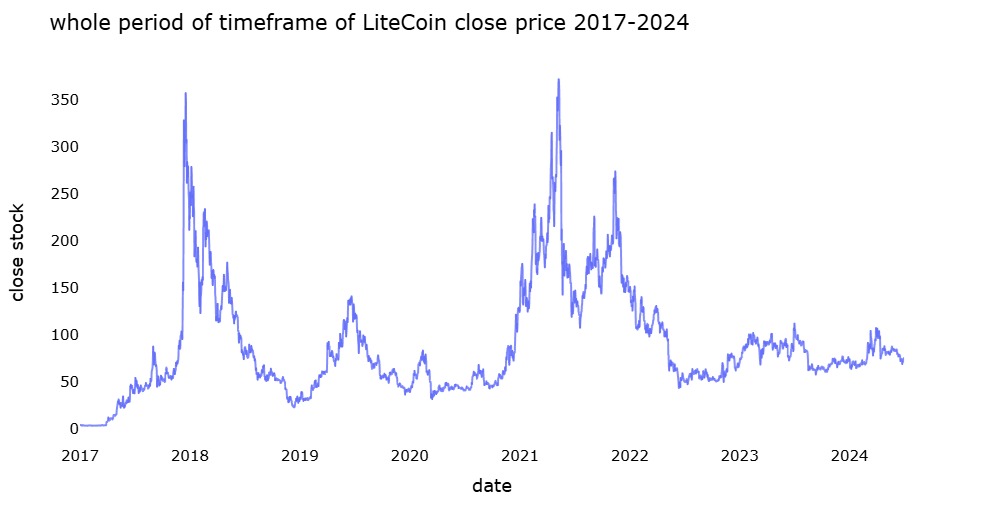
**Litecoin****Maker**

**Preprocessing data:**

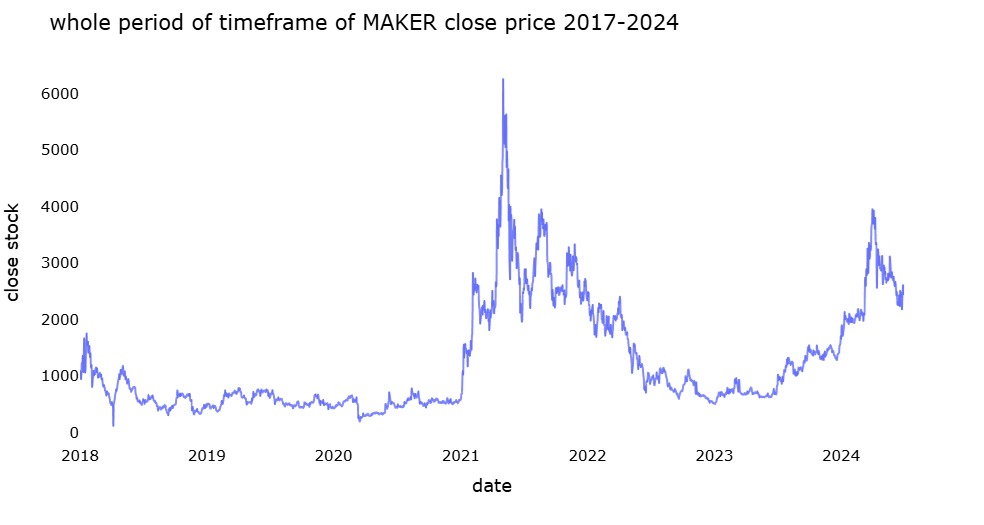
****

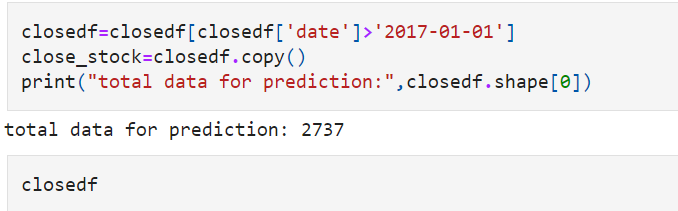
**Bitcoin****Ethereum**

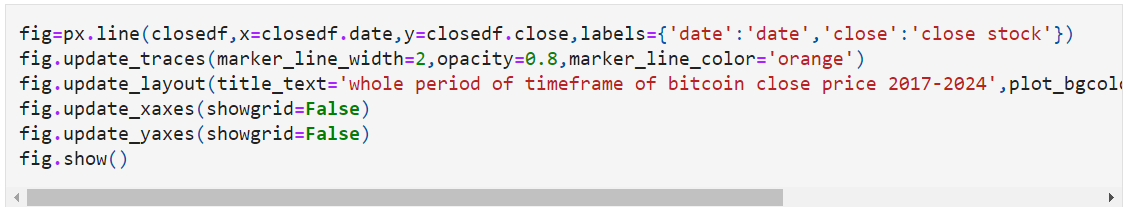
**Litecoin**

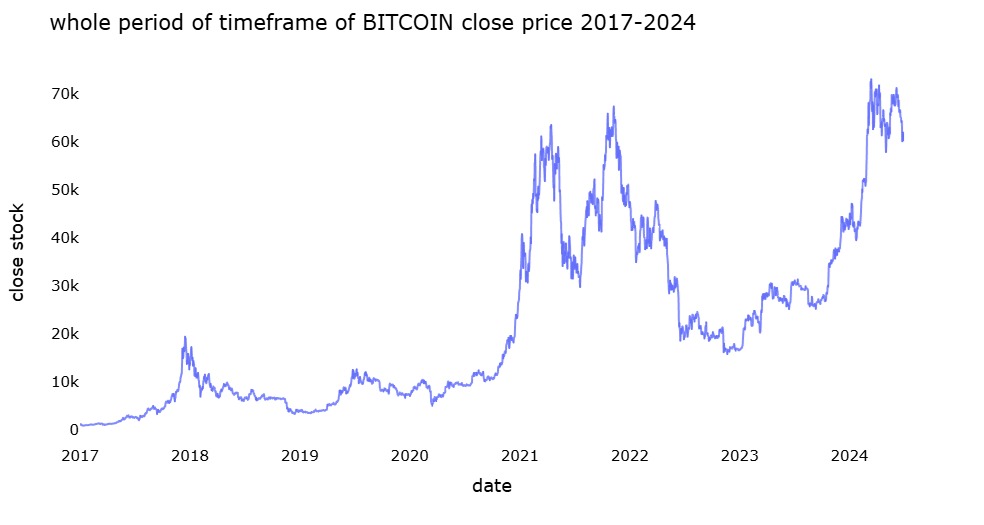


**Maker**

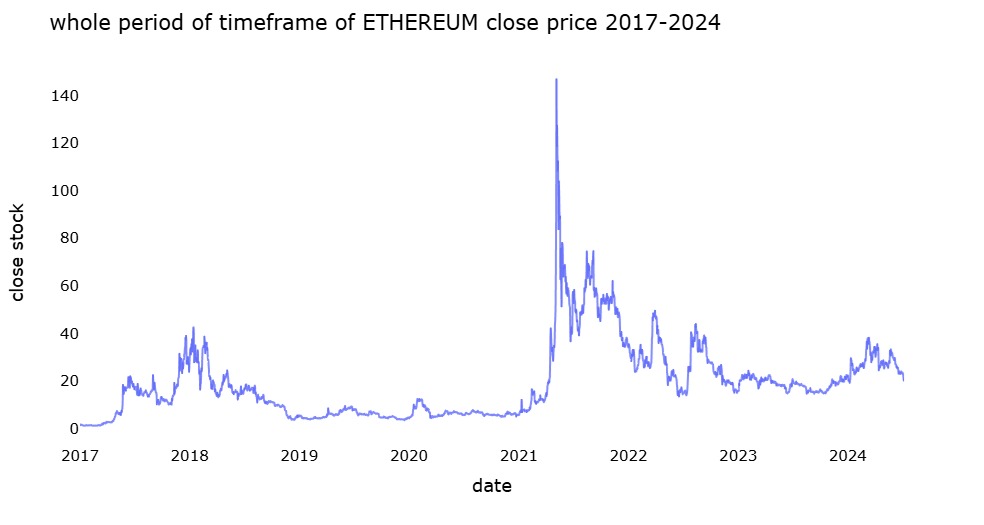


****

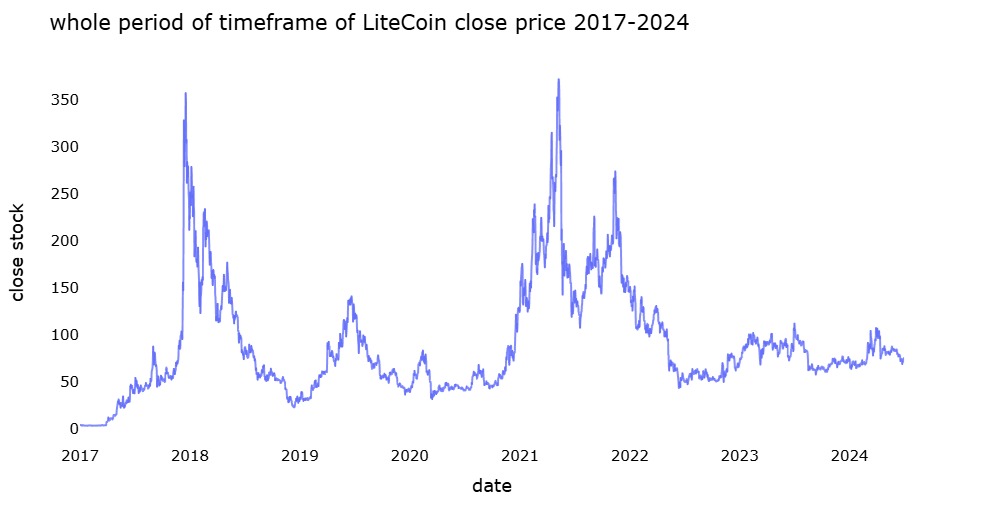
****

**Bitcoin**

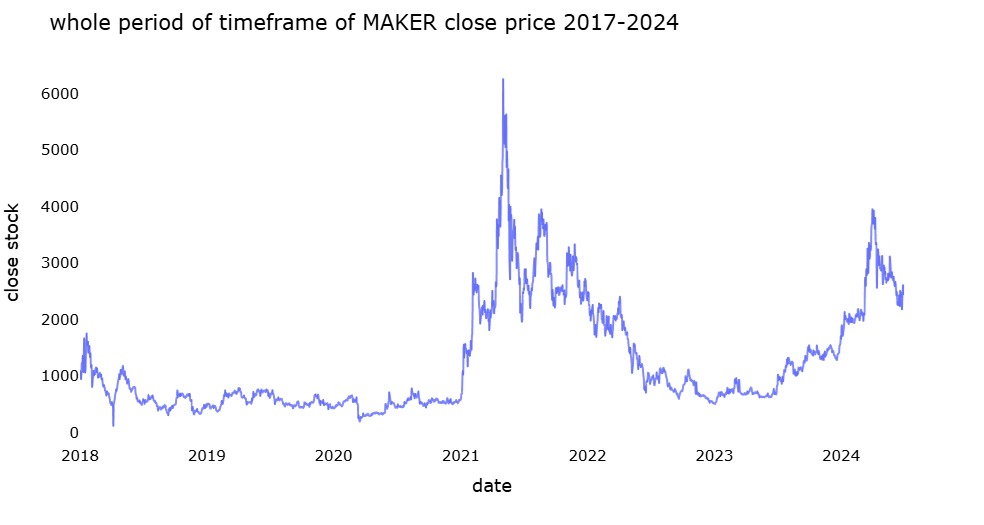
**Ethereum**



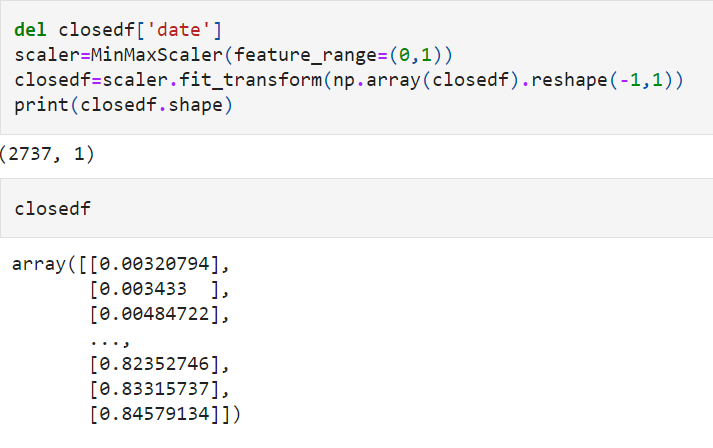
**Litecoin**



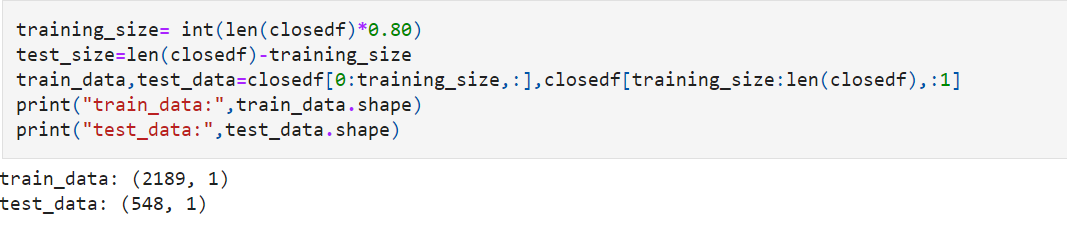
**Maker**

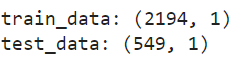




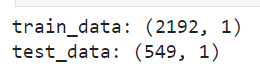
****

**Splitting data:**

**Ethereum**

****

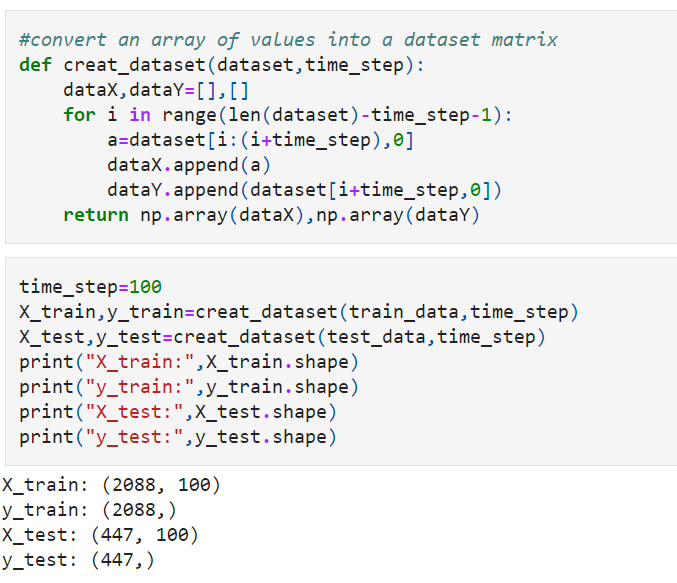
**Litecoin**

****

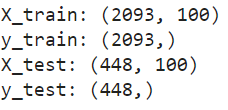
**Maker**

****

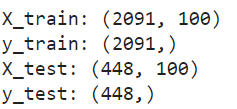
**Transforming data:**

****

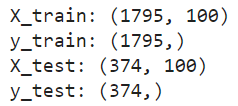
**Ethereum**

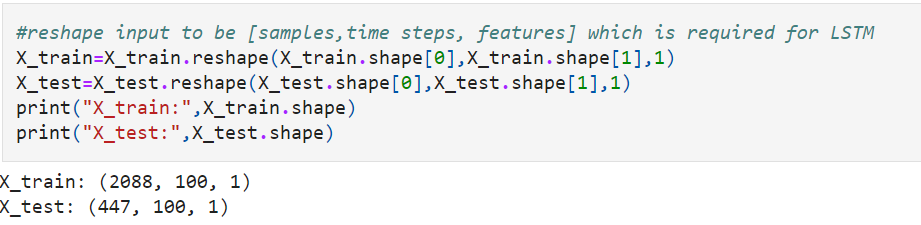
****

**Litecoin**

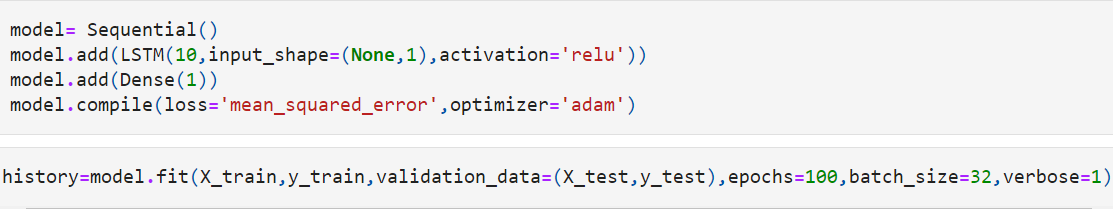
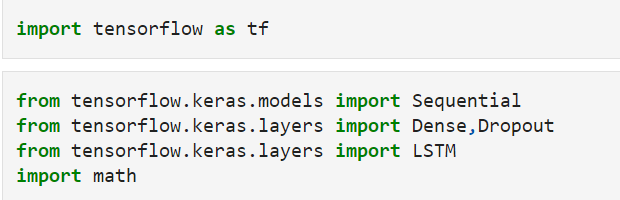
****

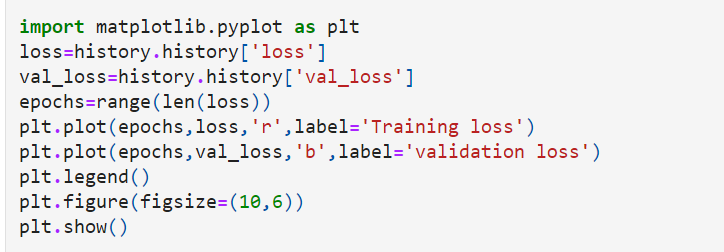
**Maker**

****

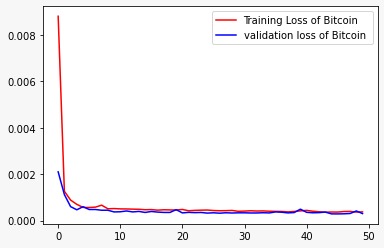
****

**Model Building:**

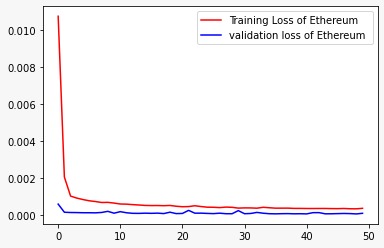
****

****

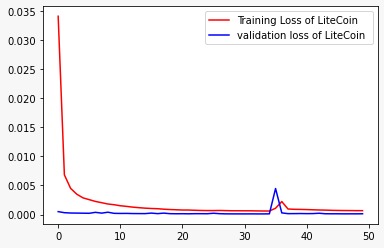
**Bitcoin**



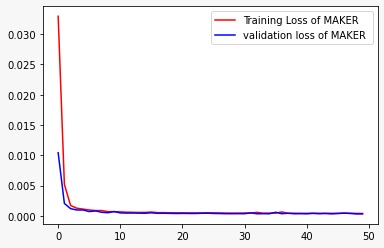
**Ethereum**



**Litecoin**

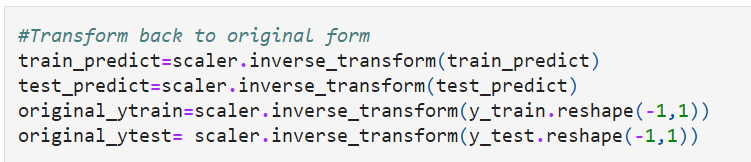


**Maker**

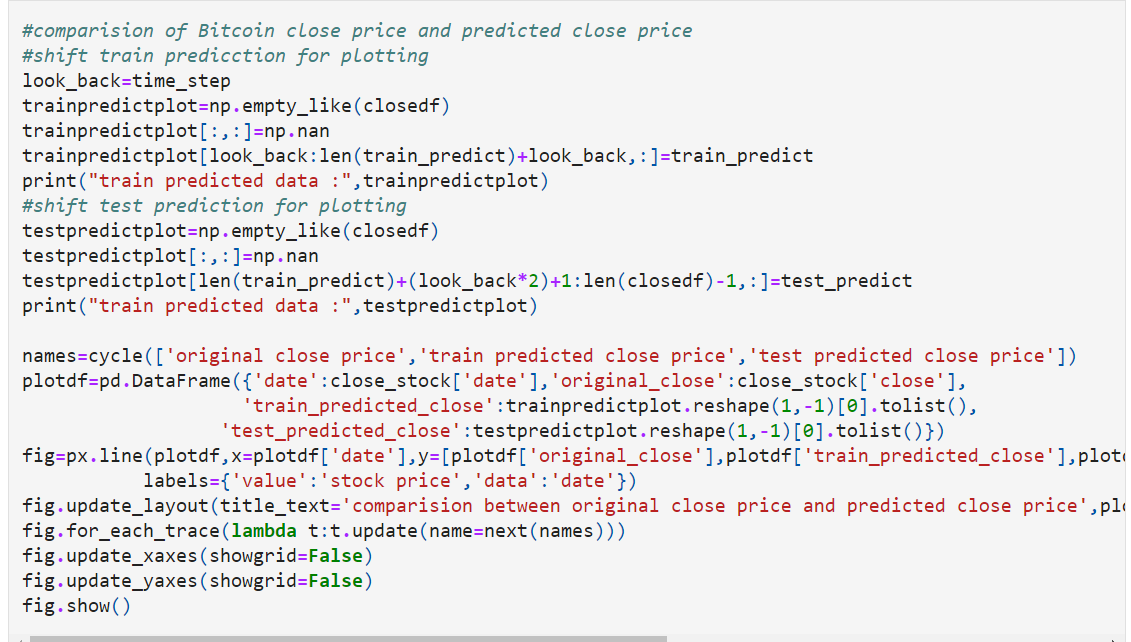


****

**Model evaluation:**

****

**Comparison of close price and predicted close price**

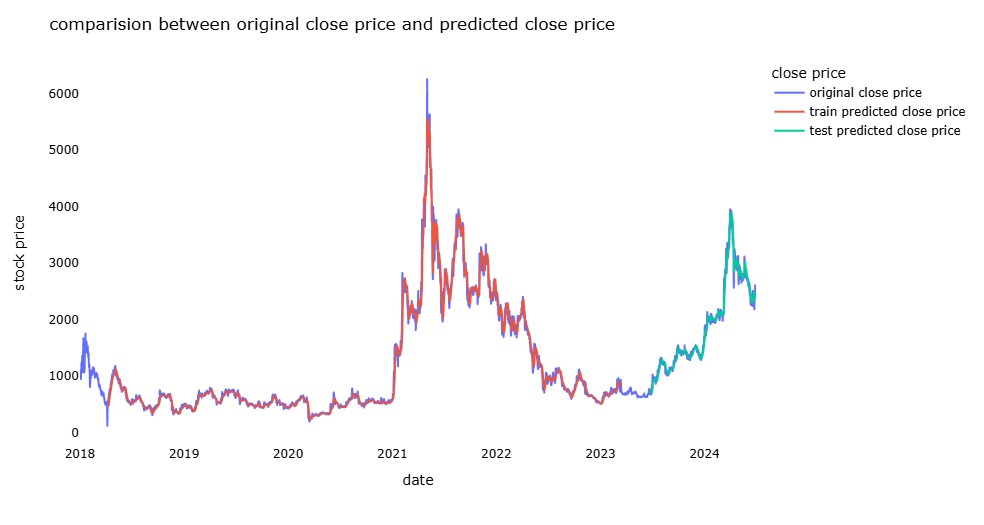
****

**Bitcoin**

**Ethereum**

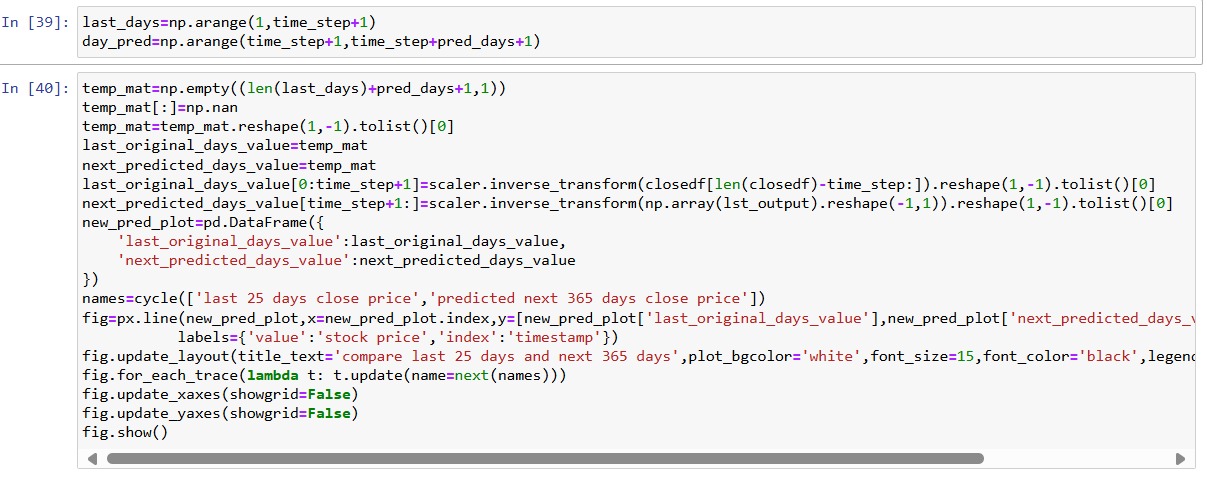
**Litecoin**

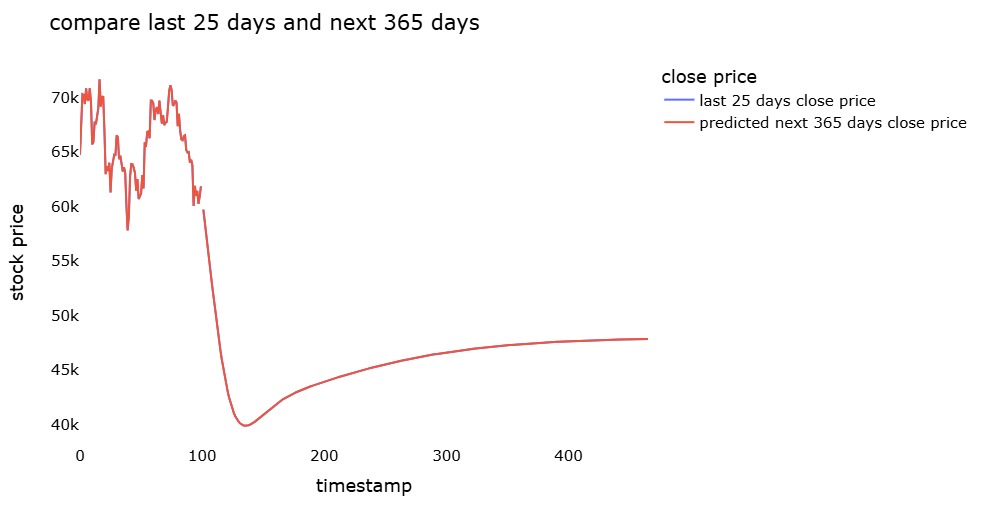


**Maker**

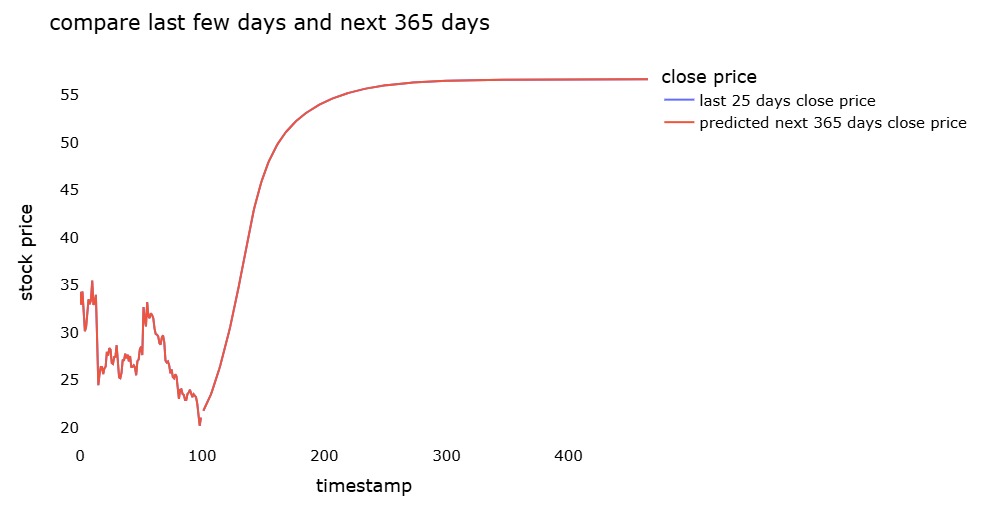
**Predicting next 365 days:**

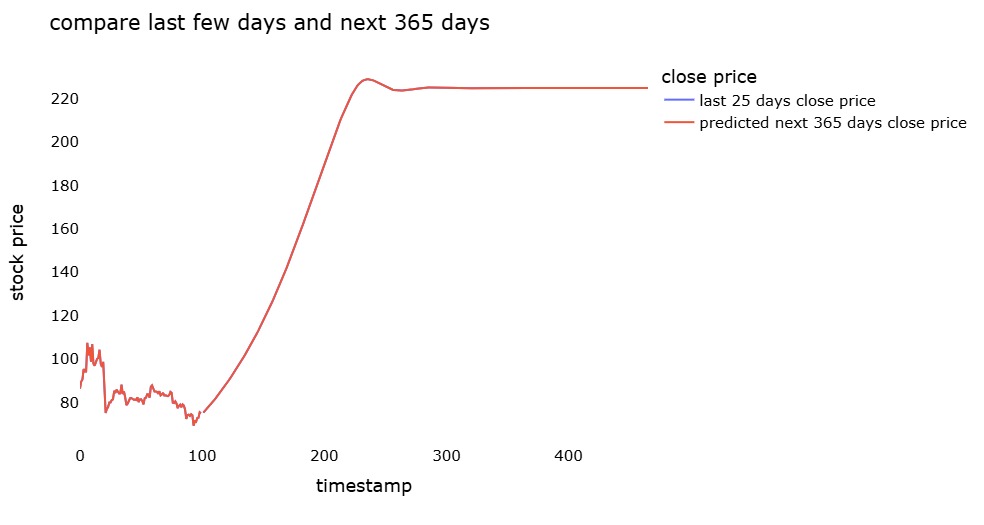


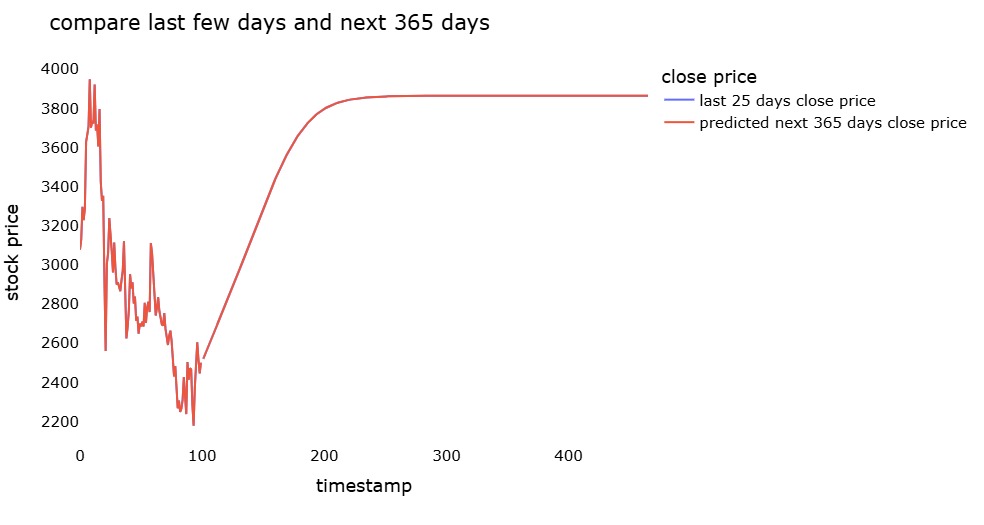


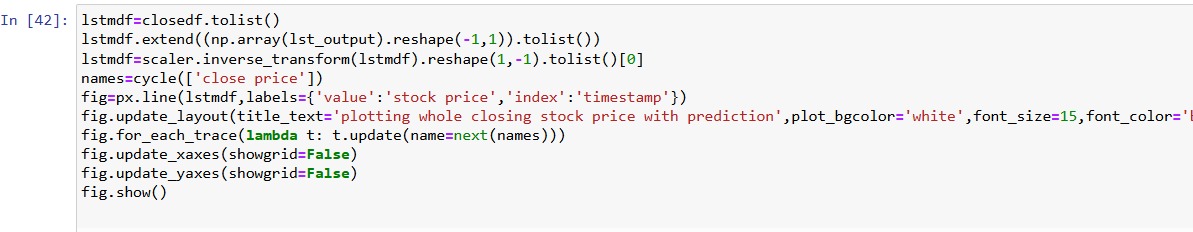
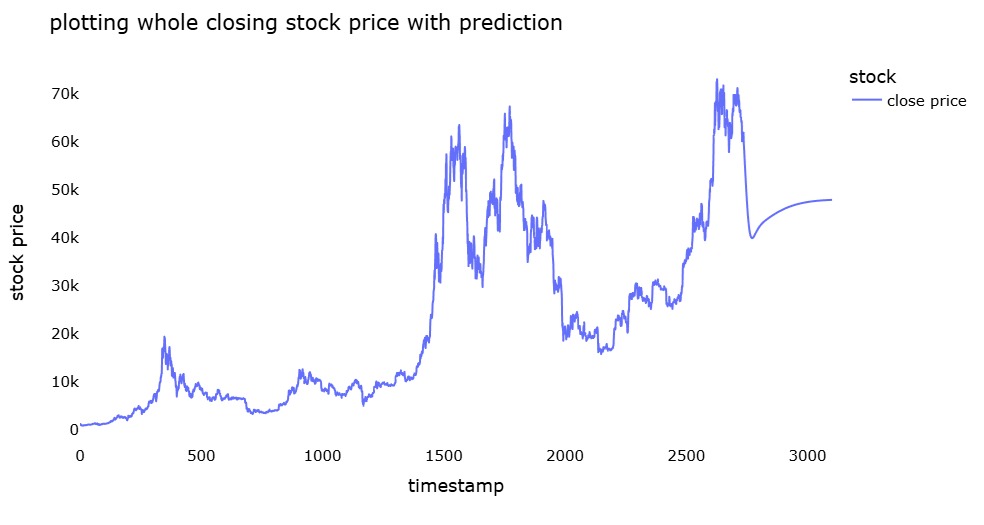
**Bitcoin**

**Ethereum**

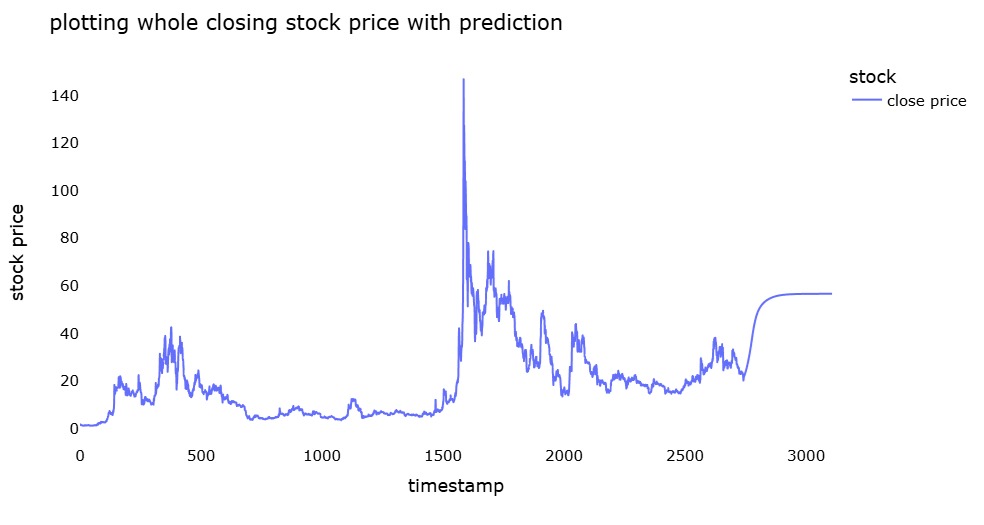


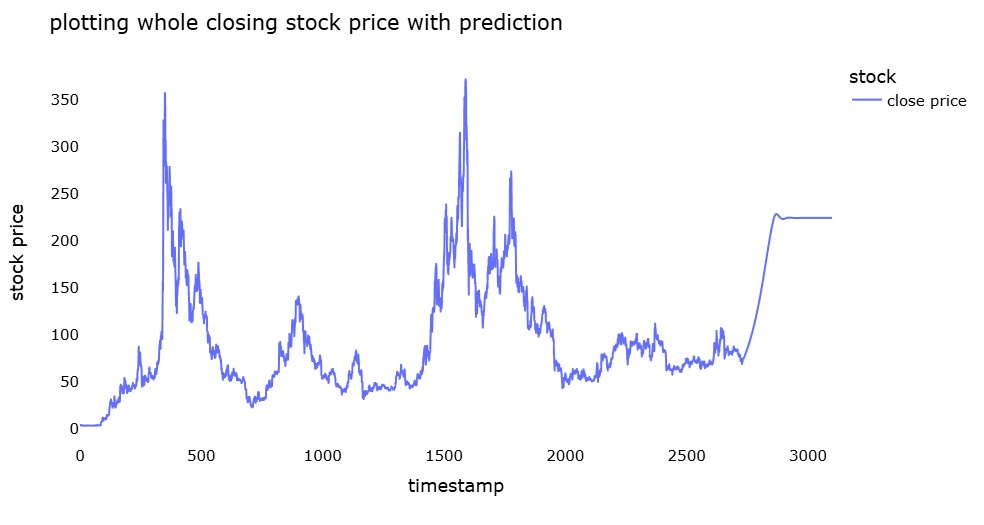
**Litecoin**

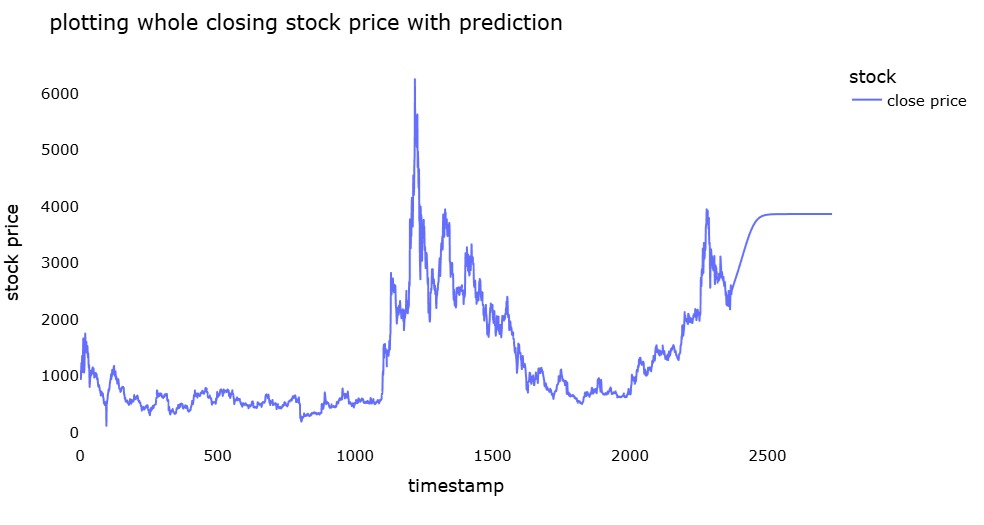
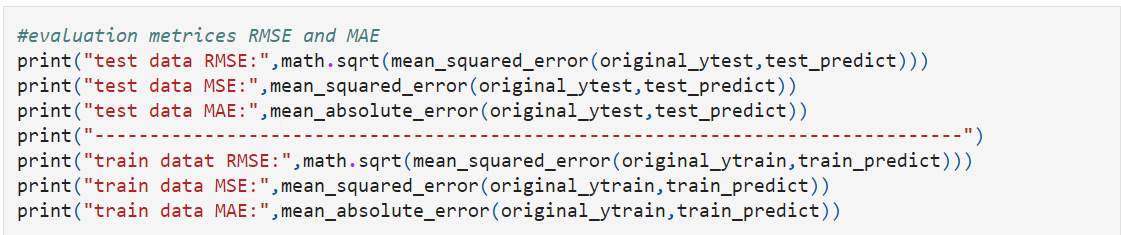
**Maker**

**Bitcoin**

**Ethereum**

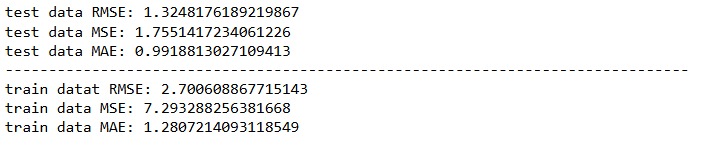


**Litecoin**

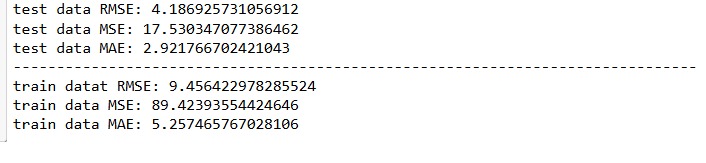
**Maker****Evaluation metricsBitcoin**



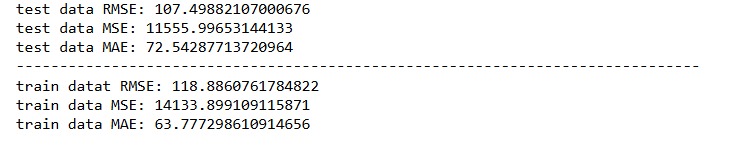
**Ethereum**



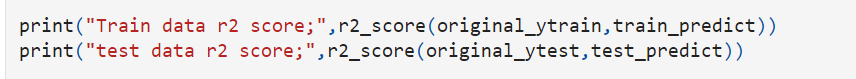
**Litecoin**



**Maker**



**R square score for regression:**

****

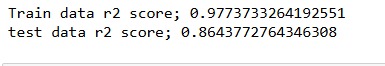
**Bitcoin**



**Ethereum**



**Litecoin**



**Maker**



**4.3 BIBILOGRAPHY**

1. [**https://en.wikipedia.org/wiki/Cryptocurrency**](https://en.wikipedia.org/wiki/Cryptocurrency)

2.[**https://www.moneycontrol.com/msite/wazirx-cryptocontrol-articles/the-journey-of-cryptocurrencies-in-india/**](https://www.moneycontrol.com/msite/wazirx-cryptocontrol-articles/the-journey-of-cryptocurrencies-in-india/)

3.[**https://bitcoin.org/en/**](https://bitcoin.org/en/)

4.[**https://en.wikipedia.org/wiki/Bitcoin**](https://en.wikipedia.org/wiki/Bitcoin)

5.[**https://en.wikipedia.org/wiki/Ethereum**](https://en.wikipedia.org/wiki/Ethereum)

6.[**https://en.wikipedia.org/wiki/Litecoin**](https://en.wikipedia.org/wiki/Litecoin)

7.[**https://en.wikipedia.org/wiki/Zcash**](https://en.wikipedia.org/wiki/Zcash)

8.[**https://kriptomat.io/cryptocurrency-prices/maker-mkr-price/what-is/#:~:text=Maker%20(MKR)%20is%20an%20ERC,tracks%20the%20price%20of%20USD**](https://kriptomat.io/cryptocurrency-prices/maker-mkr-price/what-is/#:~:text=Maker%20(MKR)%20is%20an%20ERC,tracks%20the%20price%20of%20USD)

9.[**https://docs.makerdao.com/**](https://docs.makerdao.com/)

10.[**http://repo.darmajaya.ac.id/4781/1/Time%20Series%20Analysis\_%20Forecasting%20and%20Control%20%28%20PDFDrive%20%29.pdf**](http://repo.darmajaya.ac.id/4781/1/Time%20Series%20Analysis_%20Forecasting%20and%20Control%20%28%20PDFDrive%20%29.pdf)

11.**https://en.wikipedia.org/wiki/Forecasting**

12.**https://en.wikipedia.org/wiki/Stationary\_process**

13.**https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322**

14.**https://en.wikipedia.org/wiki/Autocorrelation**

15.[**https://en.wikipedia.org/wiki/Autoregressive\_model**](https://en.wikipedia.org/wiki/Autoregressive_model)

16.[**https://otexts.com/fpp2/AR.html**](https://otexts.com/fpp2/AR.html)

17.**https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:~:text=An%20autoregressive%20integrated%20moving%20average%2C%20or%20ARIMA%2C%20is%20a%20statistical,values%20based%20on%20past%20values.**

18.**https://en.wikipedia.org/wiki/Autoregressive\_integrated\_moving\_average**

19.[**https://developer.ibm.com/tutorials/awb-arima-models-in-r/**](https://developer.ibm.com/tutorials/awb-arima-models-in-r/)

20.**https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/**

21.**https://en.wikipedia.org/wiki/Moving-average\_model**

22.**https://www.geeksforgeeks.org/understanding-the-moving-average-ma-in-time-series-data/**

23.**https://en.wikipedia.org/wiki/Long\_short-term\_memory#:~:text=Long%20short%2Dterm%20memory%20(LSTM,and%20other%20sequence%20learning%20methods.**

24.**https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm**

25.**https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/**

26.**https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/**

27.**https://medium.com/@prudhviraju.srivatsavaya/lstm-implementation-advantages-and-diadvantages-914a96fa0acb**

28.**https://www.investopedia.com/terms/l/line-graph.asp#:~:text=A%20line%20graph%E2%80%94also%20known,over%20a%20specified%20time%20interval.**

29.**https://en.wikipedia.org/wiki/NumPy**

30.**https://www.w3schools.com/python/numpy/numpy\_intro.asp**

31.**https://www.geeksforgeeks.org/introduction-to-pandas-in-python/**

32.**https://www.geeksforgeeks.org/python-introduction-matplotlib/**

33.**https://www.datacamp.com/tutorial/seaborn-python-tutorial**

34.**https://en.wikipedia.org/wiki/TensorFlow**

13. Applied Statistics, Parimal Mukhopadhyay