Governors State University

OPUS Open Portal to University Scholarship

All Student Theses

Student Theses

Fall 2022

Predicting the Price of Cryptocurrency Using Machine Learning Algorithm

Ayesha Meenaz
Governors State University

Follow this and additional works at: https://opus.govst.edu/theses

Recommended Citation

Meenaz, Ayesha, "Predicting the Price of Cryptocurrency Using Machine Learning Algorithm" (2022). *All Student Theses*. 132.
https://opus.govst.edu/theses/132

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the Governors State Computer Science Department
This Thesis is brought to you for free and open access by the Student Theses at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Student Theses by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

PREDICTING THE PRICE OF CRYPTOCURRENCY USING MACHINE LEARNING ALGORITHM

By

Ayesha Meenaz
B.Tech in Information Technology
Jawaharlal Nehru Technological University, 2015

GRADUATE CAPSTONE THESIS

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484
2022

Acknowledgment

I would like to express my gratitude to Dr. Soon-Ok Park, who has served as my mentor and chairperson throughout the project, giving me advice and offering suggestions. I wouldn't have been able to complete my research without her guidance and input. She has been a tremendous inspiration and has been very supportive. In addition, I wish to thank Professors Aslam Shahid and Freddie Kato for being members of my committee and supporting me throughout the process. It would not have been possible for me to begin my research without them. It is greatly appreciated that they have dedicated their time and energy to this project, even though they were not required to do so. I would like to extend a sincere thank you to Dr. Carrington, Division Chair, for allowing me to enroll in this program. I would like to extend a very special appreciation to Nancy Rios, Department Secretary, for all her assistance, support, and guidance throughout my graduate studies. As a final note, I would like to thank my parents for their support and encouragement throughout my entire life. As a support system, they have always been there to support me in my endeavors, and they have somehow managed to tolerate my countless late-night projects. There are no words to describe how grateful I am for where they have taken me today and what they have given me.

Abstract

It is proposed to conduct a project aimed at forecasting cryptocurrency price values. The concept of cryptocurrencies refers to computerized money that is used for a variety of transactions as well as for long-term investments. The most common cryptocurrency that most of the systems use to conduct their transactions is the Ethereum cryptocurrency. However, it needs to be noted that there are many other well-known crypto currencies other than ethereum as well. We propose to use Machine Learning for this project, which will be trained from the available cryptocurrency price data, to gain intelligence, and then use this knowledge to make accurate predictions. Trading cryptocurrency prices is one of the most popular exchanges right now. It is suggested that both day traders and investors can benefit greatly from using the suggested approach.

Table of Contents

List of Figures

1. Cryptocurrency:

Cryptocurrencies (also known as cryptos) are digital currencies that serve as a medium of exchange. Cryptography is used to secure and verify transactions and to control the creation of new units of a digital currency. Blockchain technology is the basis for many cryptocurrencies, which is a distributed ledger enforced by a distributed network of computers. In contrast to fiat currencies like the United States dollar or the British pound, cryptocurrencies do not have a central authority, making them potentially impervious to government manipulation.[2]

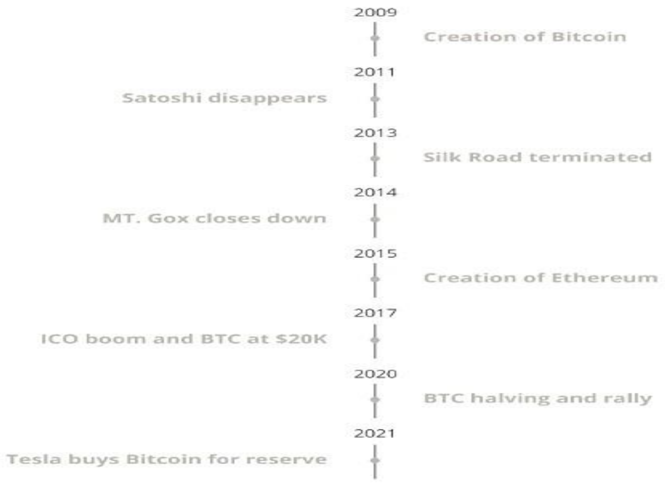


Fig 1: Crypto-related events

Cryptocurrencies like Bitcoin, Ethereum and other cryptocurrencies are powered by blockchain technology, which makes them highly appealing and functional. Blockchain is basically a network of linked blocks or an online ledger as its name suggests. A set of transactions is independently verified by each member of the network during every block. It is almost impossible to forge transaction histories since each node must verify every new block generated before it is confirmed. A copy of the online ledger must be agreed upon by the entire network, or by each individual node, or computer. Experts say blockchain technology can serve a variety of industries, including supply chains and online voting and crowdfunding. In order to decrease transaction costs by streamlining the payment process, financial institutions like JPMorgan Chase & Co. (JPM) are using blockchain technology as a means of reducing transaction costs.[3]

It is the purpose of the present study to determine whether the price of Ethereum can be predicted based on other stock market tickers similar to currency prices. A key part of this decision will be whether Ethereum can also be used as a payment medium. In order to determine if ethereum can be used as an investment or not, we will look at factors such as this. Our objectives will therefore be augmented by this. In addition, we will examine how sentiments affect Ethereum and whether they are similar to how sentiments affect stock markets. This will lead to comparisons between how ethereum and stocks perform. A central ledger was planned and built to keep track of all transactions taking place in order to ensure cryptocurrency is not recklessly spent or to keep cryptocurrency security and practicality. The Bitcoin and Ethereum blockchain is used as a means of tracking all Bitcoin and Ethereum transactions that occur throughout the world. Each block contains a hash of the previous block, so a chain of blocks is implemented. As well as maintaining the cryptocurrency, there are nodes running the network.[4]

1.1. Ethereum:

The Ethereum platform is a decentralized virtual machine that uses ETH (also known as Ether) as a means of paying transaction fees (or "gas"). Developers can use Ethereum to run decentralized applications (dApps) and create new crypto assets, such as Ethereum tokens, through the Ethereum network.

In terms of market capitalization, Ethereum is the second most valuable cryptocurrency after Bitcoin. It is also a decentralized computing platform that can run a wide variety of applications, including a universe of decentralized financial applications and services with a decentralized and decentralized architecture. There are already a number of financial tools and games running on the Ethereum blockchain, as well as complex databases on this platform. As the nonprofit Ethereum Foundation notes: “Ethereum has the potential to be more than just a payment system. It's a marketplace where you can shop for financial services, games, apps, and so on without worrying about your data being stolen or censored.” The Ethereum platform is based on the key innovation, "smart contracts," a technology that enables the development of applications based on its blockchain platform. Just as with a regular paper contract, a smart contract outlines the terms of the arrangement between two parties without needing to know who is on the other side of the deal — and without the need for an intermediary. However, unlike a traditional contract, a smart contract automatically executes when the terms are met without requiring a third party to act as an intermediary. It was in 2013 that a 19-year-old programmer (and co-founder of Bitcoin Magazine) named Vitalik Buterin released a whitepaper proposing a very flexible blockchain that would be capable of supporting almost any type of transaction. Vitalik along with a team of cofounders, including Gavin Wood, sold

$18 million in prelaunch tokens in order to crowdfund the development of the Ethereum protocol. The first public version of the Ethereum blockchain was launched in July 2015, and smart contracts began to be implemented on the Ethereum blockchain from that point onward.[5]

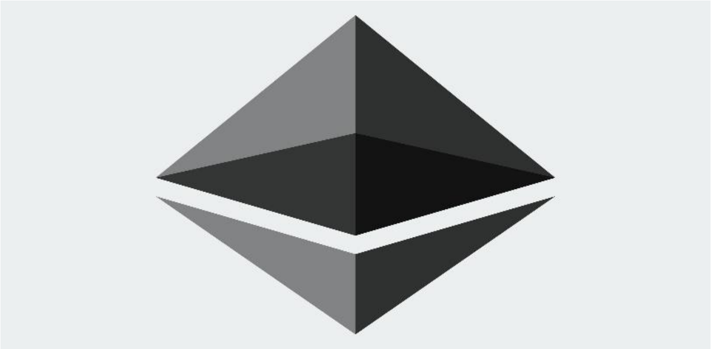


Fig 2. Ethereum logo

1.2. History:

Since the birth of the cryptocurrency industry in 2008, several figures have had a significant impact upon it. Satoshi Nakamoto is best known for the creation of Bitcoin (BTC), but Vitalik Buterin, also known for building Ethereum (ETH), has had a significant impact on the cryptocurrency movement in the past few years. A number of additional tokens have been created on the Ethereum network since its inception, known as ERC-20 tokens.

In the early days of Bitcoin, Jed McCaleb played a key role in spreading the industry's prominence by launching Mt. Gox, a place where Bitcoin transactions were frequently conducted despite its original purpose being a place where Magic: The Gathering fans could come together. When, however, the platform collapsed in 2014, it became notorious for its inability to function properly.

One of the founders of Binance, Changpeng Zhao increased the availability of crypto-assets, which has grown into one of the largest crypto exchanges today. It is my opinion that Sam Bankman-Fried is another important figure in the crypto space who impacts trading, decentralized finance (DeFi) and other aspects of the industry. He is the co-creator of FTX, the digital asset trading platform.[2]

1.3. Where can I buy cryptocurrency and how do I purchase it?

It is possible for any investor to purchase cryptocurrency from any exchange or app, such as Coinbase, Cash App, or through a broker, depending on which exchange or app they choose. The other popular way to invest in cryptocurrencies is to buy financial derivatives, including CME's Bitcoin futures, as well as other instruments like Bitcoin trusts and Bitcoin exchange-traded funds.[3]

2. Data Science:

The field of data science combines mathematics and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise in order to uncover actionable insights within an organization's data. With the help of these insights, strategic planning and decision making can be guided in a more effective manner.

Increasing amounts of data sources, and the subsequent data that they contain, has made the field of data science one of the fastest growing industries in the world today. Due to this reason, it is no wonder that the role of a data scientist has been deemed the "sexiest job of the 21st century" by Harvard Business Review. Organizations are increasingly depending on them for interpretation of data and facilitating actionable recommendations to improve business outcomes.

2.1. Lifecycle:

The data science lifecycle includes a variety of roles, tools, and processes that enable analysts to gain actionable insights. Generally, a data science project follows the following stages:

- ❖ **Importing data:** This lifecycle begins with the collection of raw data from all relevant sources--both structured and unstructured. Data can be entered manually, scraped from the web, and streamed real-time. Structured data sources include customer data, as well as unstructured data sources like log files, audio, video, pictures, the Internet of Things (IoT), and social media.
- ❖ **Store and process data:** Different storage systems need to be considered according to the type of data to be captured, since data can come in a variety of formats and structures. As a result of data management teams setting standards around data storage and structure, analytics, machine learning, and deep learning models can be carried out more efficiently. A data integration process such as ETL (extract, transform, load) jobs involves cleaning, deduplicating, transforming and combining the data. Preparing data for a data warehouse, data lake, or other repository is essential for promoting data quality.
- ❖ **Statistical analysis:** As part of this process, data scientists examine biases, patterns, ranges, and distributions within the data to generate hypotheses for a/b testing. Using Statistical analytics exploration, analysts can determine if the data is relevant for predictive analytics,

machine learning, and/or deep learning modeling. The accuracy of a model can contribute to organizations becoming reliant on them for business decision making, allowing them to achieve greater scalability.

- **Assemble insights:** Finally, insights are presented as reports and other data visualizations in order to help decision-makers and business analysts better understand them—and their impact on business. There are components in data science programming languages such as R or Python that are capable of generating visualizations, or data scientists can use specific visualization tools for their purposes.

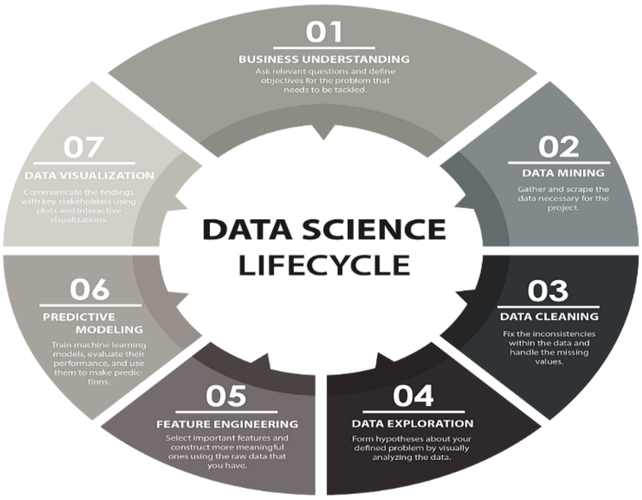

DATA SCIENCE LIFECYCLE
01
BUSINESS UNDERSTANDING
Ask relevant questions and define objectives for the problem that needs to be tackled.
02
DATA MINING
Gather and scrape the data necessary for the project.
03
DATA CLEANING
Fix the inconsistencies within the data and handle the missing values.
04
DATA EXPLORATION
Form hypotheses about your defined problem by visually analyzing the data.
05
FEATURE ENGINEERING
Select important features and construct more meaningful ones using the raw data that you have.
06
PREDICTIVE MODELING
Train machine learning models, evaluate their performance, and use them to make predictions.
07
DATA VISUALIZATION
Communicate the findings with key stakeholders using plots and interactive visualizations.

Fig 3. The Life Cycle of Data Science

2.2. Data Science Tools:

A data scientist's work is facilitated by the use of popular programming languages to perform exploratory data analysis and statistical regression. These open source tools make available prebuilt statistical models, machine learning tools, and graphics capabilities. Examples are as follows:

- **R Studio:**

This is an open source programming language and environment for developing statistical computing and graphics applications.

- **Python:**

A Python program can be written at any level and is extremely flexible and dynamic. It includes numerous libraries, such as NumPy, Pandas, and Matplotlib to help analyze data quickly and easily.

The Jupyter notebooks and GitHub are two of the most popular options for sharing code and other information among data scientists.

It's also imperative that data scientists gain proficiency in the use of big data processing platforms, such as Apache Spark, the open source framework Apache Hadoop, as well as NoSQL databases. Furthermore, they are skilled at using a variety of tools for visualizing data, including simple graphics that come with many business presentations, spreadsheets, and spreadsheet programs (such as Microsoft Excel), as well as the commercial visualization tools such as Tableau and IBM Cognos that are built specifically for this purpose, as well as open source tools such as RAW Graphs and D3.js (a library that allows you to create interactive data visualizations using JavaScript). When it comes to building machine learning models, data scientists often turn to several frameworks such as PyTorch, TensorFlow, MXNet, and Spark MLib in order to build the models.

It is well known that data science is a steep learning curve, which is why many companies are looking for ways to accelerate their return on investment for artificial intelligence projects. However, they often have difficulty finding the right talent to realize the full potential of their AI projects. To address this gap, they are turning to multipersona data science and machine learning (DSML) platforms, giving rise to a new role of “citizen data scientist”. The Multipersona DSML (Directory Services Markup Language) platform uses automation, self-service portals, and low-code/no-code user interfaces in order to help people with little to no knowledge of digital technology and no expertise in data science and machine learning create value for their businesses. A multipersona DSML platform can also be used by experts in data science to support a more technical interface. It encourages collaboration across an organization when you use it.[6]

3. Machine Learning:

With the advent of Machine Learning in the last two decades, it has become a major component of information technology and with that, one of the most central aspects of our lives, despite the fact that it is usually hidden from view. There is a growing body of evidence that smart data analysis will become even more ubiquitous as a necessary component of technological advancement as more and more data becomes available as more and more data becomes available.[7]

The concept of machine learning (ML) describes the use of mathematical models of data to teach a computer to learn without direct instruction, which is considered a subset of artificial intelligence (AI). The purpose of machine learning is to identify patterns within data using algorithms, and then using those patterns in order to create a predictive data model. As more data and experience are gathered, machine learning is able to become more accurate, just as humans gain more experience. There are a lot of scenarios that can be addressed by machine learning, because of its adaptability, as well as the fact that it can be applied on situations where the data is constantly changing, the nature of the request or task keeps changing, or coding a solution would be impossible. In spite of the fact that machine learning is a type of predictive analytics, one of the most notable differences is that machine learning is significantly easier to implement and it is able to update as it gathers more data. Predictive analytics, on the other hand, tends to work with static datasets that must be refreshed periodically.[8]

3.1. Methods of Machine Learning:

Models of machine learning can be divided into three primary categories.

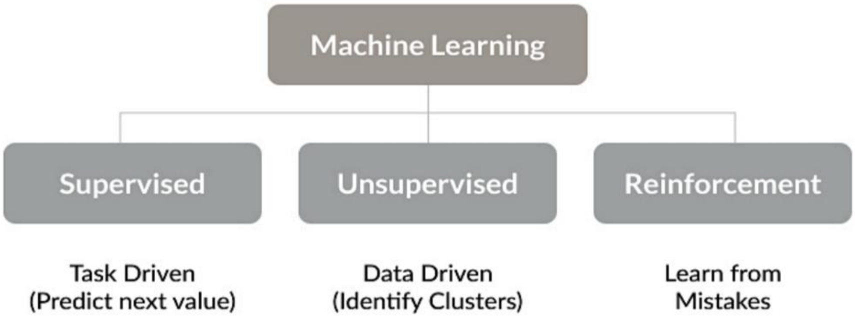


Fig 4. Types of Machine Learning

3.2. Supervised Machine Learning:

As part of supervised machine learning, algorithms are trained to classify data or predict outcomes using labeled datasets. As input data is fed into the model, the model adjusts its weights until it is properly fitted. This occurs in the cross validation process to ensure that overfitting or underfitting is avoided. A supervised learning model has the capability of helping organizations solve a wide variety of real-world problems at scale, such as categorizing spam into a separate folder from their inbox as well as other similar problems. It is possible to use neural networks, naive bayes, linear regression, logistic regression, random forest models, and support vector machines in supervised learning.[9]

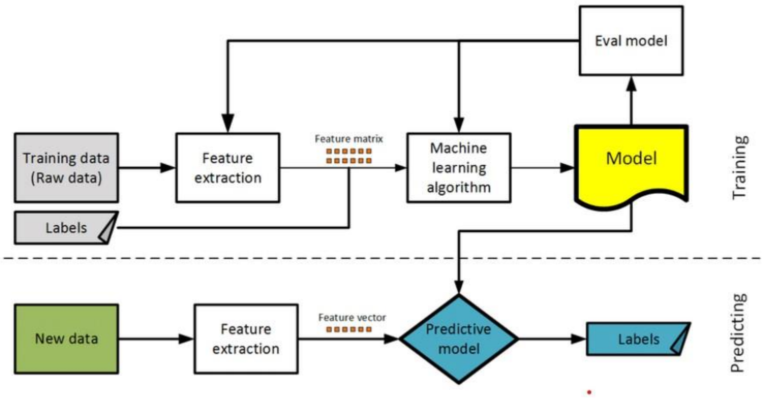


Fig 5. Supervised learning model Flowchart

The two types of supervised learning problems in data mining are classification and regression:

- **Classification:** Using a classification algorithm, test data can be accurately classified into different categories, such as apples from oranges, using an algorithm. The supervised learning algorithms, in the real world, can be used to classify spam into a separate folder from your inbox, as well. The classification algorithms that are most commonly used include linear classifiers, support vector machines, decision trees, and random forests.
- **Regression:** An algorithm that is used to understand the relationship between dependent and independent variables is known as regression in supervised learning. For example, regression models can be used for predicting numerical values based on a

number of data points, such as revenue projections for a company. There are several popular regression algorithms that have been developed over the years, such as linear regression, logistic regression and polynomial regression.[10]

It is useful to know which algorithms are currently being used for supervised learning, and the top ones are as follows:

- Linear regression
- Logistic regression
- Decision trees
- Polynomial regression
- Random forest
- K-nearest neighbors
- Naive Bayes[11]

3.3. Unsupervised Machine Learning:

In unsupervised learning, also called unsupervised machine learning, algorithms are employed to analyze and cluster unlabeled datasets using machine learning algorithms. By using these algorithms, hidden patterns can be discovered or data clusters can be discovered without the need for human input. The fact that this method is capable of finding similarities and differences in information makes it an excellent tool for exploratory data analysis, cross-selling strategies, customer segmentation, and identifying patterns and images. Moreover, principal component analysis (PCA) and singular value decomposition (SVD) are two common methods for reducing the number of features in a model through the process of dimensionality reduction. The use of neural networks, k-means clustering, and probabilistic clustering are other methods used in unsupervised learning that can also be used in unsupervised learning.[9]

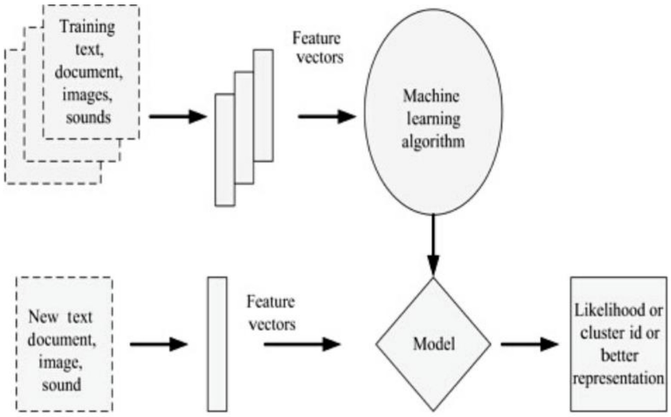

Training text, document, images, sounds
Feature vectors
Machine learning algorithm
New text document, image, sound
Feature vectors
Model
Likelihood or cluster id or better representation

Fig 6. Unsupervised Machine Learning

In unsupervised learning, there are three main tasks: clustering, association, and dimensionality reduction:

- **Clustering:** The clustering technique is a method of data mining that groups unlabeled data based on their similarities and differences. In K-means clustering, similar data points are grouped into groups, and the K value corresponds to the granularity and size of groupings. This technique is beneficial for market segmentation, image compression, etc.
- **Association:** A second type of unsupervised learning method is association learning, which is a method of finding relationships between variables in a given dataset using different rules. In addition to being used for market basket analysis and recommendation engines, such methods are also frequently employed to suggest items to customers based on the fact that people who purchased this item also purchased other items.
- **Dimensionality reduction:** An effective way to reduce the number of features (or dimensions) in a dataset is to use dimension reduction. With this technique, the number of inputs in the dataset is reduced to a manageable size while preserving the integrity of the data. It is commonly used when preprocessing data, such as when auto-encoders remove noise from visual data to improve picture quality, when using this technique. Unsupervised learning algorithms include:[10]

- Partial least squares
- Fuzzy means

- Singular value decomposition
- K-means clustering
- Apriori
- Hierarchical clustering
- Principal component analysis[11]

Semi-Supervised learning: A semi-supervised learning approach offers a compromise between supervised learning and unsupervised learning. It guides classification and feature extraction using a smaller labeled set during training. If there is insufficient labeled data for a supervised learning algorithm, semi-supervised learning can provide a solution. It may also be useful when labeling sufficient data is too expensive.[9]

3.3.1. Supervised Learning v/s Unsupervised Learning:

- A supervised learning algorithm uses labeled input and output data, while an unsupervised learning algorithm does not.
- As the algorithm "learns" from the training dataset, iteratively predicts the data and adjusts to find the correct answer. It takes human intervention up front to label the data appropriately in supervised learning models, despite their greater accuracy. According to the time of day, the weather conditions, etc., a supervised learning model can predict how long your commute will take. As the algorithm "learns" from the training dataset, iteratively predicts the data and adjusts to find the correct answer. It takes human intervention up front to label the data appropriately in supervised learning models, despite their greater accuracy. According to the time of day, the weather conditions, etc., a supervised learning model can predict how long your commute will take.
- It is important to note that unsupervised learning models still require some human intervention for validating output variables since they still work on their own to discover the inherent structure of unlabeled data. Online shoppers often purchase multiple products in groups. To validate that grouping baby clothes with diapers, applesauce, sippy cups and applesauce makes sense, a data analyst would need to confirm.[10]

3.4. Reinforcement machine learning:

It is a machine learning method that is similar to supervised learning, but the algorithm is not trained using sample data. It uses trial and error to learn as it goes. To develop a recommendation or policy for a given problem, a sequence of successful outcomes will be reinforced, so that a sequence of successful outcomes is reinforced.[9]

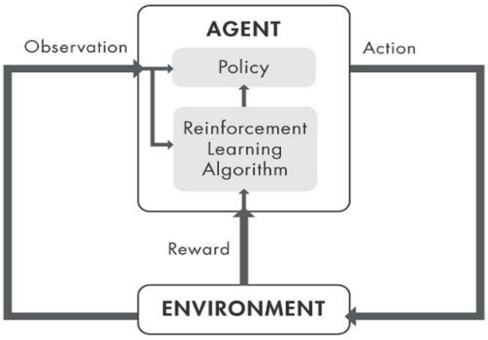


Fig 7. Reinforcement Machine Learning

3.5. Algorithms commonly used in machine learning:

- **Neural networks:** A neural network simulates the functioning of the human brain by linking hundreds of processing nodes together. In addition to recognizing patterns, neural networks play a significant role in the translation of natural language, recognition of images, speech recognition, and creation of images.
- **Linear regression:** In this technique, numerical values are predicted based on a linear relationship between different values. For example, house prices could be predicted based on historical data.
- **Logistic regression:** Using this algorithm, predictions can be made for categorical response variables, such as yes/no answers to questions. This algorithm may be beneficial for applications such as spam classification and quality control.
- **Clustering**: The clustering algorithms are based on unsupervised learning and can identify patterns in data to group it. The computer can make data scientists' lives easier by identifying differences between data items they might have overlooked otherwise.
- **Decision trees:** A decision tree is a series of linked decisions that can be represented by a tree diagram, and it can be used to predict numerical values (regression) as well as classify data into categories. A decision tree has the advantage of being easy to validate and audit, unlike neural networks, which are a black box.

- **Random forests:** A random forest is a machine learning algorithm that combines the results of a number of decision trees to figure out what value or category will be predicted.[9]

3.6. Practical uses of machine learning:

- **Automatic speech recognition:**

Automatic speech recognition (ASR), also known as computer speech recognition (CSR), or speech-to-text, is a technology that uses natural language processing (NLP) in order to translate human speech into a written document. As part of their systems, many smartphones integrate speech recognition into their systems for conducting voice searches—e.g. Siri—as well as making texting easier.

- **The customer service:**

We are experiencing a shift in customer engagement across websites and social media platforms as online chatbots replace human agents and change the way we think about customer engagement. The chatbot can answer the most frequently asked questions (FAQs) about shipping, provide users with customized advice, cross-sell products, or suggest sizes.

- **Computer Vision:**

It is possible for computers to derive meaningful information from images, videos, and other visual inputs and then take appropriate action based on this AI technology. Computer vision is a form of artificial intelligence (AI) that is powered by convolutional neural networks. It has applications in tagging social media posts, radiology imaging in healthcare, and self-driving cars.

- **Investing in stocks using automated methods:**

In order to optimize stock portfolios, AI-driven high-frequency trading platforms have been designed to perform thousands, even millions of trades per day without the need for human intervention.

- **Fraud Detection:**

Using machine learning, banks and other financial institutions can spot suspicious transactions. Through supervised learning, a model can be trained using information about fraudulent transactions known to them. Anomaly detection can identify transactions that appear atypical and should be investigated further.[11]

3.7. The top machine learning software tools:

The following are the most popular Machine Learning Software that are available on the market.

i. **Scikit-learn:** Using Scikit-Learn, one can develop machine learning applications using Python. It provides a library for the language.

Specifications:

- Analyzing and mining data is made easier with this tool.
- This package includes models and algorithms for classifying, regressing, clustering, reducing dimensions, selecting models, and preprocessing data.

The advantages are:

- It is provided with easy-to-understand documentation.
- It is possible to change parameters for any algorithm while a specific object is being called.

Cost/Plan Details: This tool is free of charge.

Official Website: scikit-learn

ii. **PyTorch:** PyTorch is a Python machine learning library that is based on the Torch computing framework, scripting language, and machine learning library that is based on Lua.

Specifications:

- Using the Autograd Module, it helps build neural networks.
- It offers a variety of neural network optimization algorithms.
- A cloud platform can be used with PyTorch.
- The platform provides distributed training, various tools, and a library of resources.

The advantages are:

- Computational graphs can be created with it.
- Using a hybrid front-end makes it easier to use.

Cost/Plan Details: This tool is free of charge.

Official Website: Pytorch

iii. **TensorFlow:** The TensorFlow library provides APIs for building and training machine learning models in JavaScript.

Specifications:

- Makes it easier for you to train and build your models.
- By using TensorFlow.js, which is a model converter, you will be able to run existing models with the help of your current models.
- This helps in the formation of neural networks.

The advantages are:

- There are two ways in which you can use it, i.e., by using script tags or by using NPM to install it.
- Using this method, it is even possible to estimate the pose of a human.

The disadvantage is: It can be challenging to learn something new.

Cost/Plan Details: This tool is free of charge.

Official Website: Tensorflow

iv. **Collab:** With Google Colab, you'll be able to build your machine learning applications using PyTorch, Keras, TensorFlow, and OpenCV libraries. You can create a machine learning application using any combination of Python, Keras, or TensorFlow.

Specifications:

- The tool is helpful in the education of machine learning.
- Provides assistance with research related to machine learning.

The advantage is:

- It can be accessed from your Google Drive.

Cost/Plan Details: This tool is free of charge.

Official Website: Colab

v. **Weka:** Using these machine learning algorithms, you can perform data mining on your data.

Specifications:

- Preparation of the data
- The classification

- A regression analysis
- A cluster analysis
- The visualization of data
- The mining association.

The advantages are:

- Offers online training courses that can be accessed at any time.
- It provides easy-to-understand, easy-to-use algorithms.
- In addition, it provides a useful tool for students.

The Disadvantage is: This product doesn't have a lot of documentation and online support.

Cost/Plan Details: This tool is free of charge.

Official Website: Waikato-weka

vi. **Keras.io:** A Keras API can be used to build neural networks. It can be used to do quick research on neural networks and is written in Python.

Specifications:

- This is a great tool for prototyping in a fast and easy way.
- A convolutional network can be created with it.
- Networks that are recurrent can be assisted by it.
- It is capable of supporting the combination of two networks at the same time.
- A CPU or GPU can be used to run it.

The advantages are:

- Convenient and easy-to-use
- Adaptable
- Widely applicable

The Disadvantage is:

- A Keras model needs TensorFlow, Theano, or CNTK.

Cost/Plan Details: This tool is free of charge.

Official Website: Keras[12]

4. Deep Learning:

The field of deep learning is a subset of machine learning, which is a subset of artificial intelligence. Artificial intelligence is a term used to describe a number of techniques that computers can use to mimic human behavior. Deep learning is just a type of machine learning, inspired by the structure of the human brain and is a type of machine learning that makes all this possible by utilizing an algorithm that is trained on data that makes everything possible. The goal of deep learning algorithms is to derive similar conclusions to human reasoning as they continuously analyze data with a logical structure. To do this, they use neural networks that are multilayered.

A neural network is based on the structure of the human brain. Just as our brains identify patterns and classify information, neural networks can do the same thing. A neural network's layer’s act as a filter, working from gross to subtle, thereby increasing the likelihood of detecting and presenting a correct result. In deep neural networks, the brain uses the same concept of comparing new information to known objects whenever it receives it. A neural network can perform many tasks, including clustering, classification, and regression.

We can train neural networks to classify samples into different categories using a labeled data set. Alternatively, we can group and sort unlabeled data based on similarities between samples. Machine learning algorithms can perform similar tasks to neural networks (but classical algorithms cannot). Deep learning models can solve tasks that machine learning models cannot, because artificial neural networks have unique capabilities. Achieving self-driving cars, chatbots, and personal assistants like Alexa and Siri would not be possible without deep learning. All these technologies relie on neural networks, so Google Translate would remain primitive as it was ten years ago. Netflix would not know what movies to recommend.

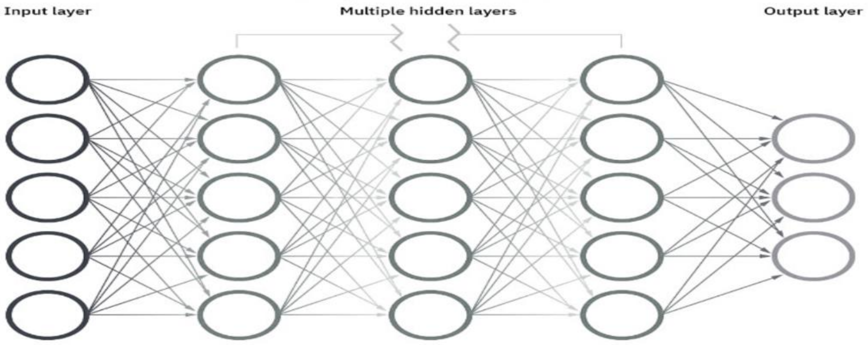


Fig 8. Deep Neural Network

4.1. Deep Learning Popularity:

A deep learning model is more powerful than a machine learning model:

4.1.1. NO EXTRACTION OF FEATURES:

One of the advantages of deep learning over machine learning is that the process of feature extraction is redundant. A number of traditional machine learning methods were utilized before deep learning was employed, including decision trees, SVMs, naive Bayes classifiers and logistic regression. These algorithms are known as flat algorithms as well. This is a technical term for algorithms that cannot be applied directly to raw data (such as. CSVs, images, text, etc.). Instead, a preprocessing step is required called feature extraction.

In classic machine learning algorithms, feature extraction results in a representation of the given raw data. For example, we can classify the data into several categories or classes. For optimal results, this preprocessing layer must be adapted, tested, and refined over several iterations. Feature extraction usually requires a detailed understanding of the problem domain, as it is usually quite complex.

The artificial neural networks that are used in deep learning do not require a feature extraction step since the layers learn a representation of the raw data directly and independently.

4.1.2. BIG DATA ERA: A New Era of Computing:

There is a second huge advantage to deep learning, and an important part of understanding why it has become so popular, is that it is powered by massive quantities of data. As big data becomes more prevalent, new innovations in deep learning will have huge breakthroughs in the near future. In contrast, traditional machine learning models such as SVM and naive Bayes classifiers do not continue to improve after a saturation point is reached, whereas deep learning models tend to improve with the increasing amount of training data.[13]

4.2. The 5 most common uses for deep learning:

4.2.1. The use of social media:

The use of deep learning can be applied to the analysis of a large number of images. This can be used by social media networks to understand more about their users to help with targeted advertisements and following suggestions.

4.2.2. The financial sector:

Deep learning uses neural networks to predict stock prices, to develop trading strategies, and to spot security threats and to prevent fraudsters from taking advantage of them.

4.2.3. The healthcare sector:

Using deep learning algorithms, healthcare workers may be able to determine what type of tests and treatments are best for their patients based on trends and behaviors. They can also use deep learning algorithms to decide what kind of tests and treatments are the most effective for their patients.

4.2.4. Cybersecurity:

With deep learning, advanced threats can be detected more effectively than traditional malware solutions because they are able to recognize new, suspicious behaviors rather than just responding to a known threat database like traditional malware solutions.

4.2.5. Digital assistants:

A number of digital assistants are examples of deep learning as they rely on natural language processing (NLP) to provide answers to questions and learn the habits of the users. These assistants include Siri, Cortana, Google Assistant, and Alexa.[14]

4.3. The use of MATLAB for Deep Learning:

MATLAB can be used to do all kinds of deep learning, from managing huge data sets to performing complex computations. It also has multiple specialized toolboxes that you can use for machine learning, neural networks, computer vision, and automated driving. You don't need to be an expert to use MATLAB's deep learning tools. Get started quickly, and create and visualize your models, as well as deploy them to servers and embedded devices, all with just a few lines of code, without learning any programming.

A major reason why MATLAB is so successful in deep learning is that it allows you to do the following:

4.3.1. Easily create and visualize models:

With MATLAB, you can easily import pretrained models and visualize and debug intermediate results as you adjust your training parameters.

4.3.2. Deep Learning Without Expertise:

As a data scientist you may find it difficult to learn about your industry or the type of application that you are working in, which makes MATLAB the perfect tool for learning about deep learning; in addition, MATLAB enables domain experts to do deep learning - rather than handing the task over to a data scientist who may not have any expertise in your area.

4.3.3. Automatically label images and videos based on ground truth:

With MATLAB, users can label objects within images interactively and can automate ground truth labeling within videos to train and test deep learning models in less time and with better results. This interactive and automated method of training and testing can lead to better results in less time.

4.3.4. Implement a single workflow for deep learning:

MATLAB offers a unique way to combine multiple domains into a single workflow. You can think and program simultaneously in MATLAB. Besides offering tools and functions for deep learning, it also provides tools and functions for a range of domains that feed into deep learning algorithms, such as signal processing, computer vision, and data analysis. As soon as the results of your deep learning models have been applied to your existing applications, you can automate the deployment of the models over enterprise systems, clusters, clouds, or embedded devices by using MATLAB.[15]

4.4. Deep Learning Vs Machine Learning:

- Basically, deep learning is a specialization of machine learning in which relevant features are extracted manually from images and used in a machine learning workflow to categorize objects in the image. A deep learning workflow helps to extract relevant features automatically from images, so that relevant features can then be used to create a model to categorize them.
- Also, deep learning is capable of performing what is called "end-to-end learning", which is when a network is given raw data and a task to do, such as classification, and it automatically learns how to perform the task.
- There are a number of differences between deep and shallow learning algorithms. The difference between deep and shallow learning algorithms is that deep learning algorithms can scale with data, whereas shallow learning algorithms converge as the amount of training data and examples increases.
- Deep learning networks have the advantage of continuously improving its performance as the data size increases, making this a very attractive feature.

- In machine learning, the feature extraction and modeling steps are done manually, while in deep learning, the steps are automated. Machine learning chooses features to sort images and then chooses a classifier.[15]

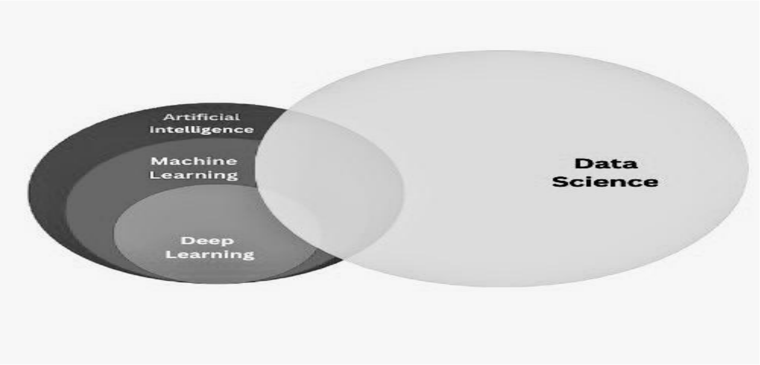


Fig 9. AI/ML/DL

5. Time Series Analysis in Machine Learning:

In machine learning, time series forecasting is a particularly important area, as a great deal of prediction problems involve a time component. Time series forecasting is of utmost importance. While time series problems add additional information, they also make them more challenging than many other prediction tasks because they have a time component.[16]

Data forecasting using time series analysis comprises the use of some significant model in order to forecast future outcomes based on known past outcomes in order to predict future conclusions.

- Time series analysis is instrumental in identifying patterns in change over time. These patterns are used to identify trends, cycles, and irregular movements within a time series, as well as identifying new trends emerging from time series.
- A proper data model is essential for forecasting accurate future sales, GDP, and global temperatures through time series in order to generate accurate forecasts, such as future sales, GDP, and global temperatures.

In addition to predicting future, Time Series can also be used for many other types of investigations, such as circadian rhythms, seasonal behaviours, trends, and changes.

The purpose of time series analysis is to record data at regular intervals of time in order to be able to make an informed decision which can lead to a positive outcome for trade, and so they have many applications, such as Stock Market and Trend Analysis, Financial Analysis and Forecasting, Inventory Analysis, Census Analysis, Yield prediction, Sales Forecasting and many more.

A time-series model is generally defined as a combination of three variables: the Autoregressive (AR), the Integrated (I), and the Moving Average (MA) as well as combinations of these variables that include Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA).

Models reflecting measurements taken at the same time are more relevant than those taken at different times.

5.1. How to implement time series analysis in machine learning?

In the world of artificial intelligence, it is well known that Machine Learning is one of the most powerful techniques in the processing of images, speech, and natural language for a vast number of elaborated datasets that are available at this time. On the other hand,

- Time series problems do not have usually interpreted datasets, despite the fact that there are numerous sources of data for these problems, as a result, they may exhibit considerable variances in terms of features, properties, attributes, temporal scales, and dimensionality of the datasets.
- It is necessary to have such sorting algorithms in time series analysis, so that it is capable of learning time-dependent patterns across multiple models which are different from images and speech.
- In order to perform machine learning tasks, real-world business applications are crucial. For instance, classification, clustering, forecasting, and anomaly detection all require real-world business applications to work.

5.2. Time-Series Forecasting Using Machine Learning Methods:

5.2.1. Univariate Time-Series Forecasting: The Univariate Time-Series Forecasting method relies on two variables in order to create its predictions, which are time and the field in which we are looking to make predictions.

Example:

One example of this would be when determining a person's heart rate per minute simply by observing previous measurements of their heart rates. In this case, two parameters are in play - time (minute) and heart rate (per minute).

5.2.2. Multivariate Time-series Forecasting: The Multivariate Time-series Forecasting method is another approach to forecasting problems, which is based on the fact that one variable will be fixed in time and all others will have various parameters.

Example:

As an example, if we were to predict the temperature of a city for the coming week, the only difference we would see is the fact that now the temperature will consider factors such as the following:

- The amount of rain that has fallen and the duration of the rain,
- Amount of humidity,
- An estimated wind speed,
- Amount of precipitation,
- In terms of atmospheric pressure, etc.,

This in turn will result in an accurate prediction of the temperature in the city. As you can see, all of these factors are closely related to temperature and contribute tremendously to it.

5.3. Time-series forecasting with machine learning models:

- ARIMA Model
- ARCH/GARCH Model
- Vector Autoregressive Model or VAR model
- LSTM (Long Short Term Memory)

In this way, we are able to handle long structures during the training process and create predictions based on the previous data.[17]

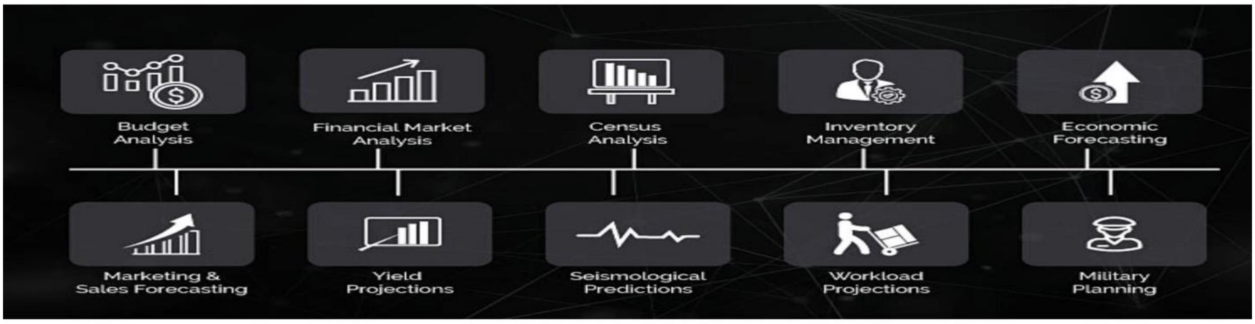


Fig 10. Applications of Time Series

6. Final Analysis:

The purpose of this research paper is to provide you with guidance on time series forecasting using machine learning, as well as how to avoid some of the pitfalls that can occur. In this paper, I will describe how one might seem to be able to develop a successful model and then decide to produce it in the real world. I want to emphasize how to evaluate the model accuracy, and how to not just rely on common error metrics like mean percentage error, R2 score, etc., can be very misleading if you are relying on the model without thinking about how to evaluate the model accuracy. In the event that they are applied without caution, they can be very misleading.

Models based on machine learning for time series forecasts:

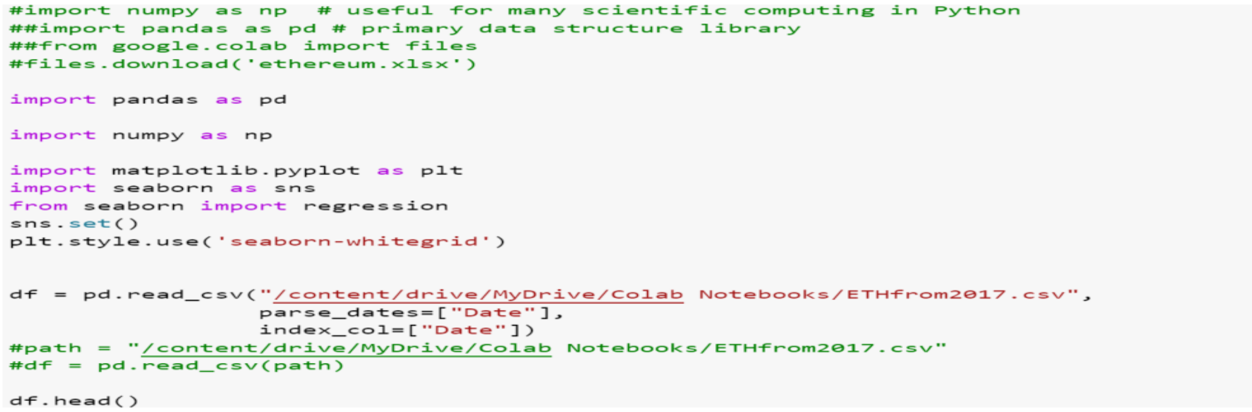
Several types of time-series forecasting models are available for use. This particular project involved me using a special type of neural network, the long short-term memory network, which is an LSTM network, which makes predictions based on the previous data in the past. However, I have observed that simpler models often provide just as accurate predictions in many cases as more complex models, which is why it is so popular for language recognition, time series analysis and so much more. In order to include temporal information in models such as the random forest, gradient boosting regression model, and time delay neural network, delays can be added to the input of the model so that the data can be represented at different points in time using models such as random forest, gradient boosting regressor, and time delay neural network. Because TDNN's (Time delay neural network) are sequential in nature, rather than implementing a recurrent network, they are implemented as feedforward neural networks.

Open source software libraries for implementing models:

There are many ways to construct neural networks, but my preference is to use Keras, as it is a high-level neural networks API written in Python that can be combined with TensorFlow, CNTK, or Theano to construct neural networks. Usually, I use Scikit-Learn as a free software machine learning library for other types of models. The Python NumPy and SciPy libraries, along with support vector machines, random forests, gradient boosting, k-means, and DBSCAN, are designed to work in tandem with support vector machines, random forests, gradient boosting, and DBSCAN in order to perform classification, regression, and clustering tasks.

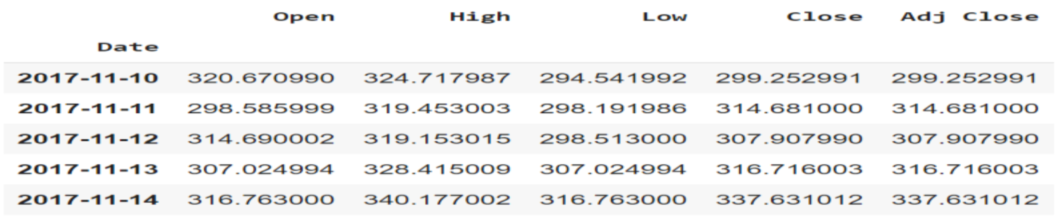
Programming:

As a first step, I loaded the libraries in my google.colab. I had a data set that I got from Yahoo.finance. The data set I was working with was from Google's website colab.research.com, so I had to use the library google.colab to load the data. The data for a time series is typically stored in spreadsheet format, in the form of a .csv file, or in another file format that contains the date and the measured value. In order to read the time series dataset (a csv file on ETH-USD) as a pandas dataframe, we will use the read_csv() function in the pandas package.



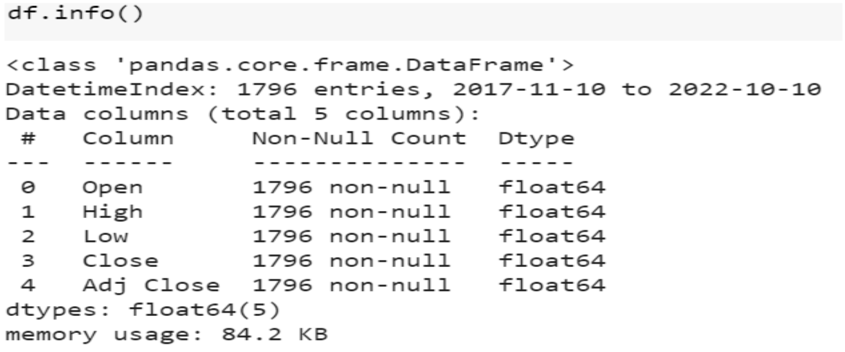
```
#import numpy as np  # useful for many scientific computing in Python
##import pandas as pd # primary data structure library
##from google.colab import files
#files.download('ethereum.xlsx')

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
from seaborn import regression
sns.set()
plt.style.use('seaborn-whitegrid')


df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/ETHfrom2017.csv",
                 parse_dates=["Date"],
                 index_col=["Date"])
#path = "/content/drive/MyDrive/Colab Notebooks/ETHfrom2017.csv"
#df = pd.read_csv(path)

df.head()
```

Fig 11: Import Data with Pandas



	Open	High	Low	Close	Adj Close
Date					
2017-11-10	320.670990	324.717987	294.541992	299.252991	299.252991
2017-11-11	298.585999	319.453003	298.191986	314.681000	314.681000
2017-11-12	314.690002	319.153015	298.513000	307.907990	307.907990
2017-11-13	307.024994	328.415009	307.024994	316.716003	316.716003
2017-11-14	316.763000	340.177002	316.763000	337.631012	337.631012

In this step, we display information about a DataFrame, including its index type and columns, as well as the non-null values and the amount of memory used. As a result, we gain an understanding of the different columns and their data types.



```
df.info()

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1796 entries, 2017-11-10 to 2022-10-10
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Open       1796 non-null   float64
 1   High       1796 non-null   float64
 2   Low        1796 non-null   float64
 3   Close      1796 non-null   float64
 4   Adj Close  1796 non-null   float64
dtypes: float64(5)
memory usage: 84.2 KB
```

The visual representation of a time series:

To visualize the series, let's use the matplotlib package with Matplotlib.

Matplotlib is a powerful Python library for creating static visualizations, animated visualizations, and interactive visualizations. Matplotlib simplifies the difficult parts of creating a visualization and makes them easier. Plot a graph from 1st Oct 2017 to 10th Oct 2022.

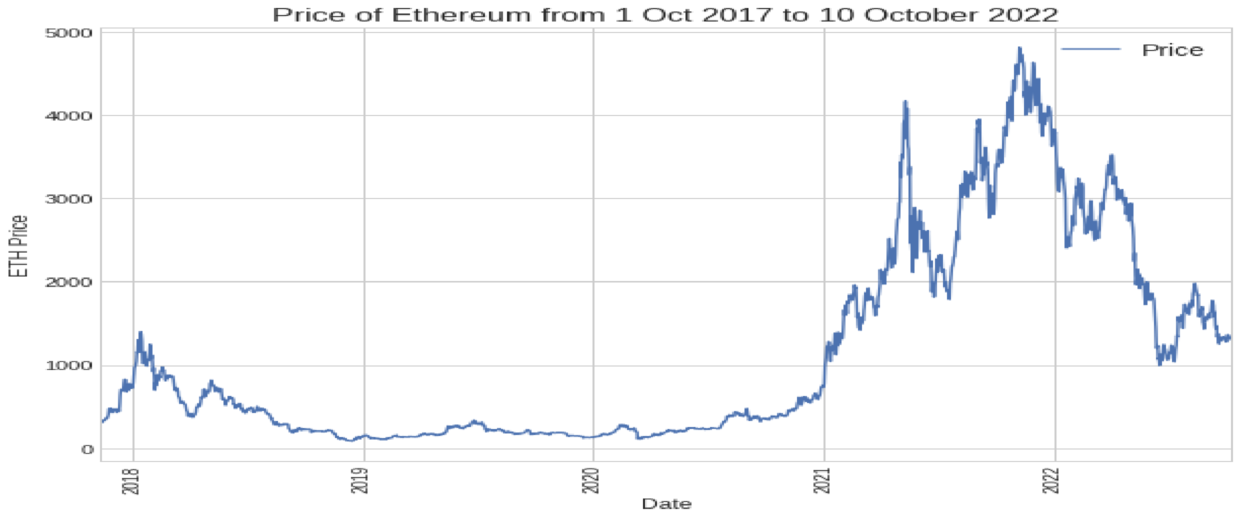
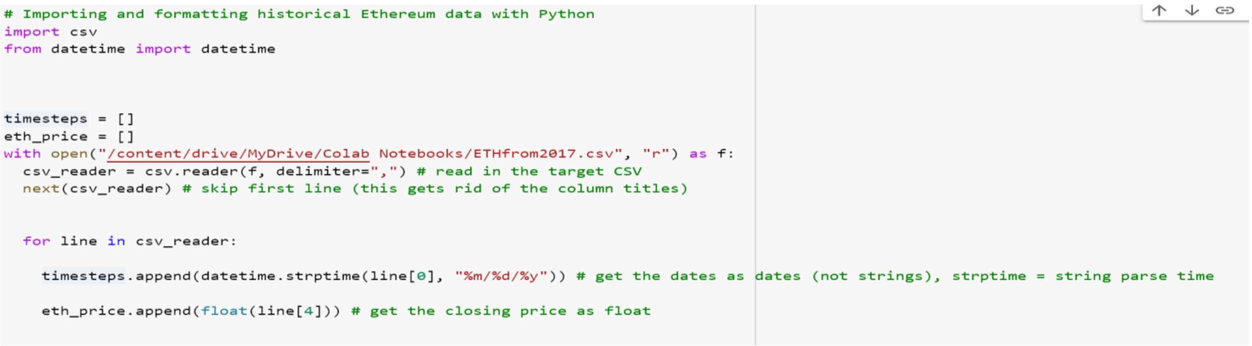


Fig 12: Price of Ethereum from 1st Oct 2017 to 10 October 2022

We are getting the price of ethereum for the period of 1st October 2017 to 10th October 2022.

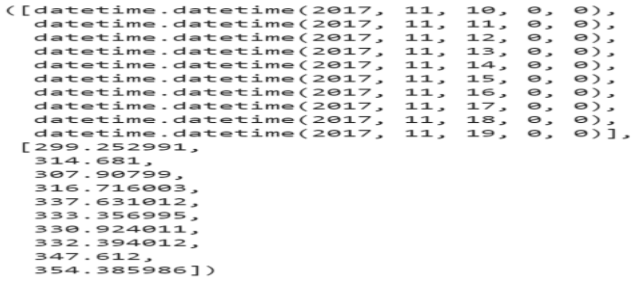
Now let us import and format historic Ethereum data in a variety of ways. Using the function "with", we load the csv file "/content/drive/MyDrive/Colab Notebooks/ETHfrom2017.csv" which takes care of the date time column and assembles it into a float value.

Dates and times do not have a data type of their own in Python, but they can be handled using a module called datetime, which is built into Python itself, so that there is no need to install it externally. Python Datetime module comes built into Python and is therefore not required. Various functions are provided to handle dates, times, and time intervals in Python's Datetime module. When you manipulate a date or datetime object in Python, you are actually manipulating an object and not a string.



```
# Importing and formatting historical Ethereum data with Python
import csv
from datetime import datetime


timesteps = []
eth_price = []
with open("/content/drive/MyDrive/Colab Notebooks/ETHfrom2017.csv", "r") as f:
  csv_reader = csv.reader(f, delimiter=",") # read in the target CSV
  next(csv_reader) # skip first line (this gets rid of the column titles)


  for line in csv_reader:

    timesteps.append(datetime.strptime(line[0], "%m/%d/%y")) # get the dates as dates (not strings), strptime = string parse time

    eth_price.append(float(line[4])) # get the closing price as float
```

Fig 13: Importing Historical Ethereum Data with Python

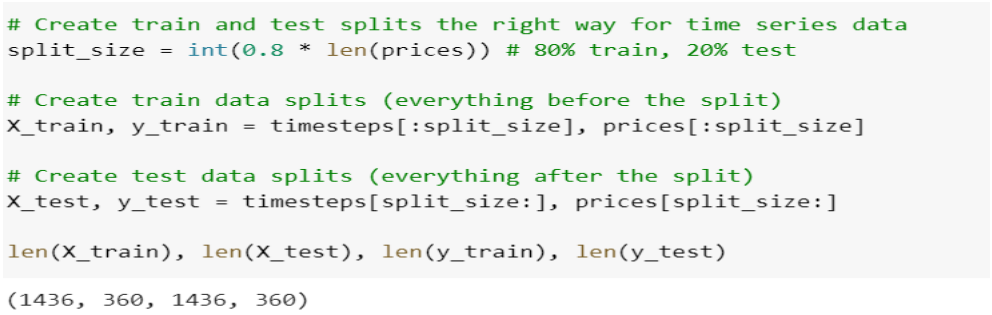


```
([datetime.datetime(2017, 11, 10, 0, 0),
  datetime.datetime(2017, 11, 11, 0, 0),
  datetime.datetime(2017, 11, 12, 0, 0),
  datetime.datetime(2017, 11, 13, 0, 0),
  datetime.datetime(2017, 11, 14, 0, 0),
  datetime.datetime(2017, 11, 15, 0, 0),
  datetime.datetime(2017, 11, 16, 0, 0),
  datetime.datetime(2017, 11, 17, 0, 0),
  datetime.datetime(2017, 11, 18, 0, 0),
  datetime.datetime(2017, 11, 19, 0, 0)],
 [299.252991,
  314.681,
  307.90799,
  316.716003,
  337.631012,
  333.356995,
  330.924011,
  332.394012,
  347.612,
  354.385986])
```

Data Splitting:

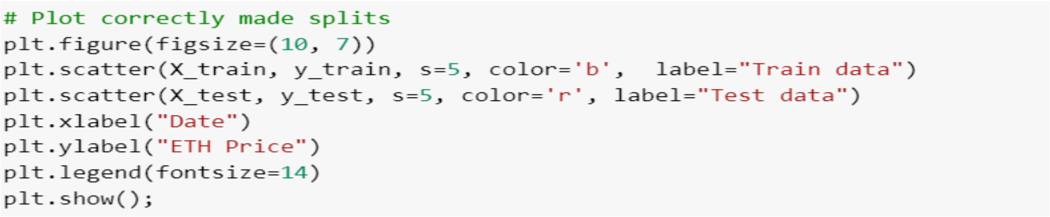
As soon as we take in some data, we make some predictions, and then we tell our machine learning model whether its predictions were good or bad. A comparison is then conducted between the predictions of the model and the labels, and the differences between the predictions and the labels are calculated using some metric, such as the mean squared error or the cross entropy, in order to assess the differences. Our model will do better at predicting our labels as we add more data to the training process and as it learns from its mistakes. Our model will perform better than if we only used 70% or 80% of the data for training. The problem is that we can no longer evaluate the true performance of our model if we use all the data.

The next step is to run the analysis on the data, and to split it into two sets - training data and test data, with 80% and 20% of the data each. Choosing the split size of training data and the length of training and test data.



```
# Create train and test splits the right way for time series data
split_size = int(0.8 * len(prices)) # 80% train, 20% test

# Create train data splits (everything before the split)
X_train, y_train = timesteps[:split_size], prices[:split_size]

# Create test data splits (everything after the split)
X_test, y_test = timesteps[split_size:], prices[split_size:]

len(X_train), len(X_test), len(y_train), len(y_test)

(1436, 360, 1436, 360)
```

Now, let's look at how the Ethereum prices in USD have changed over time by splitting using the following code to scatterplot the changes over time:



```
# Plot correctly made splits
plt.figure(figsize=(10, 7))
plt.scatter(X_train, y_train, s=5, color='b',  label="Train data")
plt.scatter(X_test, y_test, s=5, color='r', label="Test data")
plt.xlabel("Date")
plt.ylabel("ETH Price")
plt.legend(fontsize=14)
plt.show();
```

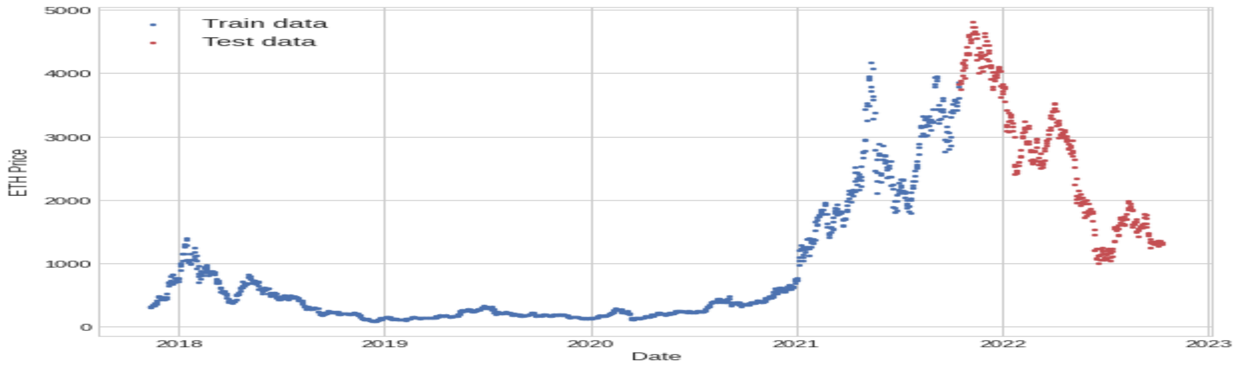


Fig 14: Plotting train (past) and test (future) split

It is interesting to note that the prices were increased in 2018 and then decreased in 2019. One interesting aspect of the price fluctuation is that it was the case that the prices were increased during the period from 2021 to 2022. The dataset under consideration cannot be generalized, since it is just a small sample that is for one year and it is hard to generalize anything with crypto currency.

Performing experiments based on a model:

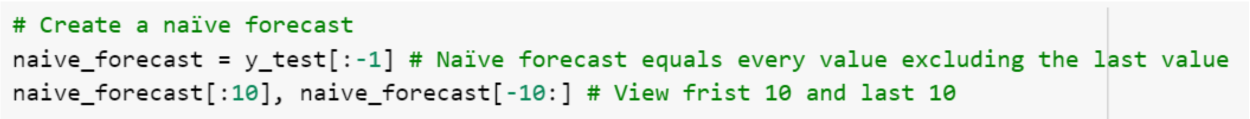
The data we collected has now been prepared, and a series of modeling experiments will be conducted to identify the best model in terms of performance.

Our method of prediction will be the naive forecast, in which all forecasts will be set to the value of the last observation since it has the lowest probability of error. Mathematically, this can be expressed as follows:

$$\hat{Y}_{T+h|T} \ = \ YT$$

This means that the prediction at timestep t is equal to the value at timestep t-1.

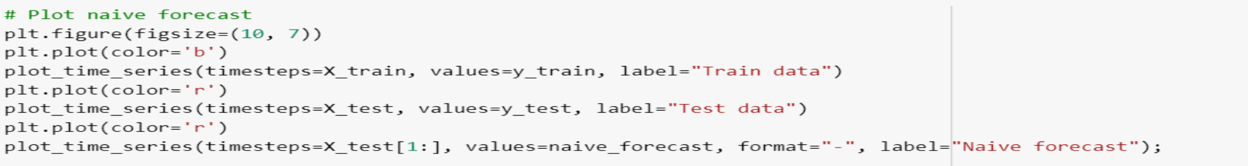
Here is how we can write this in code:



```
# Create a naïve forecast
naive_forecast = y_test[:-1] # Naïve forecast equals every value excluding the last value
naive_forecast[:10], naive_forecast[-10:] # View frist 10 and last 10
```

The naive forecast is simply predicting the previous timestep value as the next step value when the horizon is 1.

Here we plot the naive forecast predictions using the plotting method. Note that X_test needs to be offset by 1 since it will not have a value for the first time step.



```
# Plot naive forecast
plt.figure(figsize=(10, 7))
plt.plot(color='b')
plot_time_series(timesteps=X_train, values=y_train, label="Train data")
plt.plot(color='r')
plot_time_series(timesteps=X_test, values=y_test, label="Test data")
plt.plot(color='r')
plot_time_series(timesteps=X_test[1:], values=naive_forecast, format="-", label="Naive forecast");
```

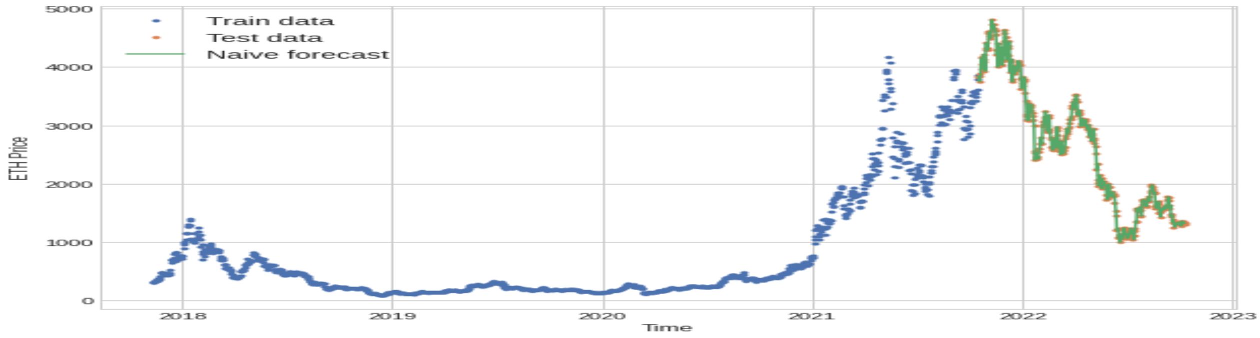


Fig 15: Plotting Naïve Forecast

We can see that the naive forecast follows the test data quite closely. As we will see building future models, it is actually quite difficult to beat this naive forecast baseline since we are not actually building a model, we are simply offsetting the test values by one index. A naive forecast predicts the next value based upon the previous value. That is, the prediction at timestep t is equal to the prediction at timestep t-1.

We can obtain the following result by offsetting the values by 300 timesteps.

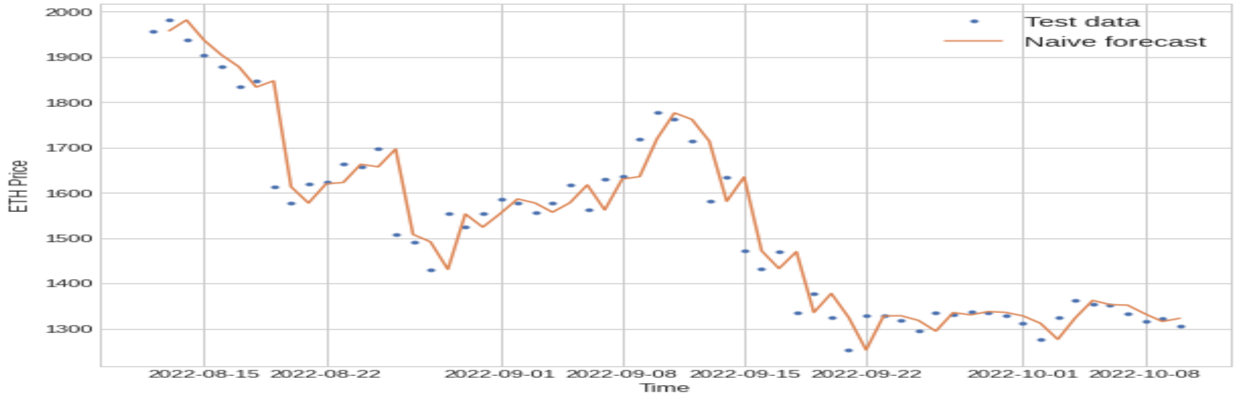


Fig 16: Closer look of the previous Naïve Forecast

Tensorflow:

The Tensorflow library is one of the most popular JavaScript libraries for Machine Learning. By using Tensorflow, we can train and deploy machine learning directly in the browser. Any Web Application can be enhanced with machine learning functions by using Tensorflow.

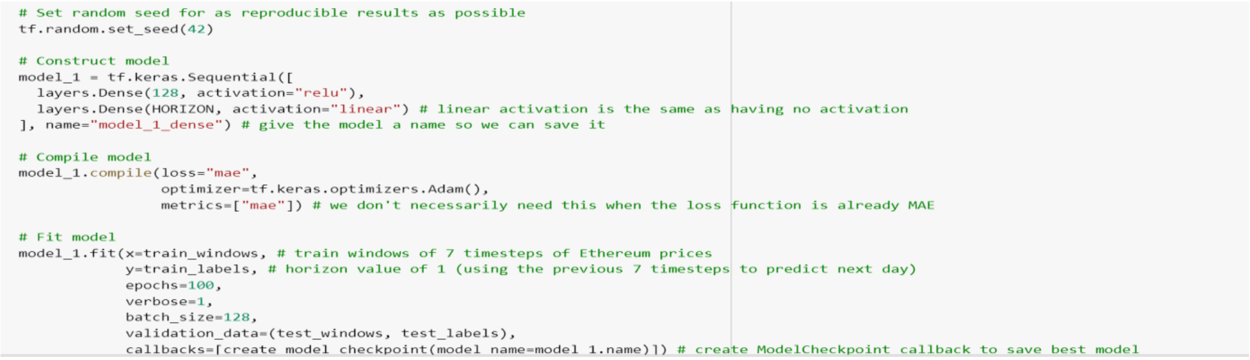
- TensorFlow uses the tensor data type as its main data type.
- Using the tf.tensor() method, you can create a Tensor from any N-dimensional array:
- The TensorFlow platform can help you implement best practices for data automation, model tracking, performance monitoring, and model retraining using best practices. In order for a product or service to be successful, it is essential that model training be automated and tracked over the lifecycle of the product, service, or business process.

I have then created a function to calculate the mean absolute scaled error where I got a value of 0.99777603 and then found the average price of Ethereum in test dataset with a value of 2618.662220116667.

Sequential API (Application Programming Interface):

Sequential API is a framework that arranges Keras layers sequentially, so it is called Sequential API because this is the core idea of Sequential API. A large number of the ANN (artificial neural network) have layers that are sequentially arranged, and data flows from one layer to another layer until it reaches the output layer at the end of the ANN.

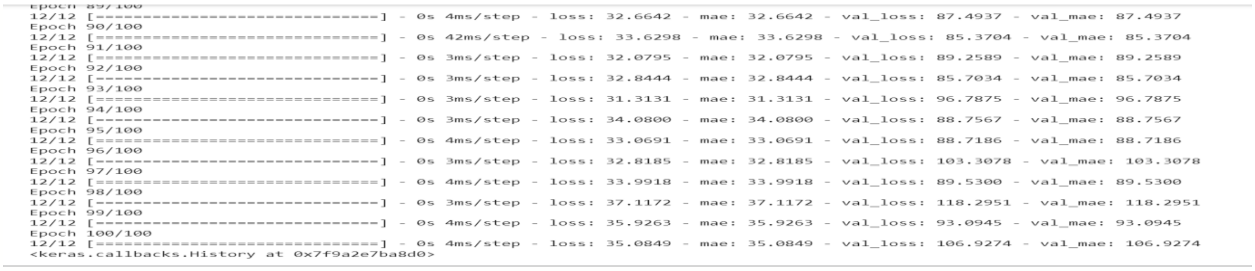
We will create the sequential model on the training data and we will be able to observe the loss for each epoch of the training data.



```
# Set random seed for as reproducible results as possible
tf.random.set_seed(42)

# Construct model
model_1 = tf.keras.Sequential([
  layers.Dense(128, activation="relu"),
  layers.Dense(HORIZON, activation="linear") # linear activation is the same as having no activation
], name="model_1_dense") # give the model a name so we can save it

# Compile model
model_1.compile(loss="mae",
                optimizer=tf.keras.optimizers.Adam(),
                metrics=["mae"]) # we don't necessarily need this when the loss function is already MAE

# Fit model
model_1.fit(x=train_windows, # train windows of 7 timesteps of Ethereum prices
            y=train_labels, # horizon value of 1 (using the previous 7 timesteps to predict next day)
            epochs=100,
            verbose=1,
            batch_size=128,
            validation_data=(test_windows, test_labels),
            callbacks=[create_model_checkpoint(model_name=model_1.name)]) # create ModelCheckpoint callback to save best model
```

Fig 17: Adjust input using tf.keras.Sequential

It can be seen that here, we have defined a sequential model using the Keras library, in which 128 neurons in the first dense layer are activated through a linear activation function, whereas 1 neuron in the second dense layer is activated through linear activation function, and so on. Using the Adam Optimizer to optimize the model result with the mean absolute error as the loss function, we will compile the model with this loss function,



```
Epoch 89/100
12/12 [==============================] - 0s 4ms/step - loss: 32.6642 - mae: 32.6642 - val_loss: 87.4937 - val_mae: 87.4937
Epoch 90/100
12/12 [==============================] - 0s 42ms/step - loss: 33.6298 - mae: 33.6298 - val_loss: 85.3704 - val_mae: 85.3704
Epoch 91/100
12/12 [==============================] - 0s 3ms/step - loss: 32.0795 - mae: 32.0795 - val_loss: 89.2589 - val_mae: 89.2589
Epoch 92/100
12/12 [==============================] - 0s 3ms/step - loss: 32.8444 - mae: 32.8444 - val_loss: 85.7034 - val_mae: 85.7034
Epoch 93/100
12/12 [==============================] - 0s 3ms/step - loss: 31.3131 - mae: 31.3131 - val_loss: 96.7875 - val_mae: 96.7875
Epoch 94/100
12/12 [==============================] - 0s 3ms/step - loss: 34.0800 - mae: 34.0800 - val_loss: 88.7567 - val_mae: 88.7567
Epoch 95/100
12/12 [==============================] - 0s 4ms/step - loss: 33.0691 - mae: 33.0691 - val_loss: 88.7186 - val_mae: 88.7186
Epoch 96/100
12/12 [==============================] - 0s 3ms/step - loss: 32.8185 - mae: 32.8185 - val_loss: 103.3078 - val_mae: 103.3078
Epoch 97/100
12/12 [==============================] - 0s 4ms/step - loss: 33.9918 - mae: 33.9918 - val_loss: 89.5300 - val_mae: 89.5300
Epoch 98/100
12/12 [==============================] - 0s 3ms/step - loss: 37.1172 - mae: 37.1172 - val_loss: 118.2951 - val_mae: 118.2951
Epoch 99/100
12/12 [==============================] - 0s 4ms/step - loss: 35.9263 - mae: 35.9263 - val_loss: 93.0945 - val_mae: 93.0945
Epoch 100/100
12/12 [==============================] - 0s 4ms/step - loss: 35.0849 - mae: 35.0849 - val_loss: 106.9274 - val_mae: 106.9274
<keras.callbacks.History at 0x7f9a2e7ba8d0>
```

Fig 18: Trained Model with 100 epochs

We ensure that the model trains well with 100 epochs and a batch size of 128, and we also pass the test data so that it can be validated. As we can see, the loss decreases with each epoch, resulting in a mean absolute error of 35.0849, and when the model is evaluated, we find a mean absolute error of 106.9274.

In order to evaluate the best performing model_1 on test data, load in the saved best performing model_1. In this function, we use a model to make predictions on input_data and return a 1D array containing the predictions. We make predictions based on model_1 on the test dataset, view the results, and evaluate the predictions.

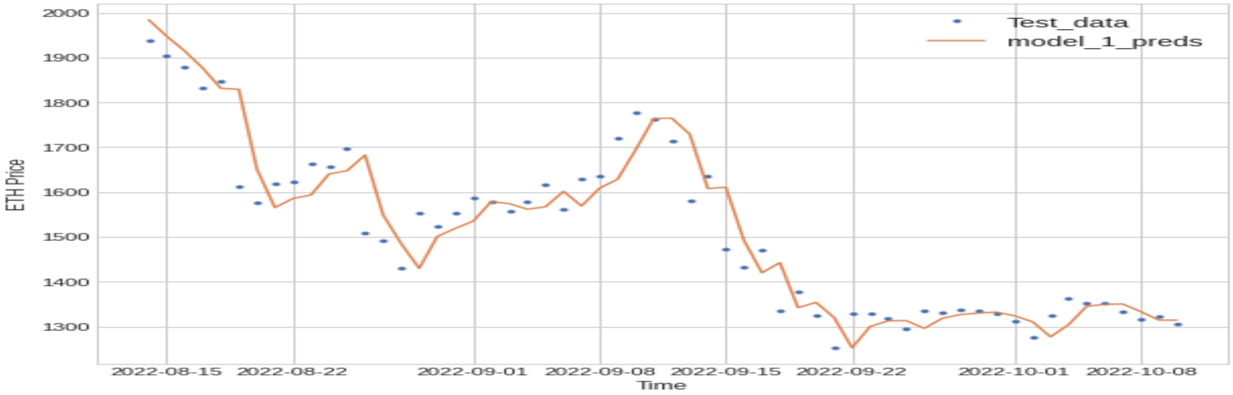


Fig 19: Model-1 Predictions

Plotting the data and comparing the test data with the prediction of model_2 s using the horizon(1) and window size(30) dimensions.

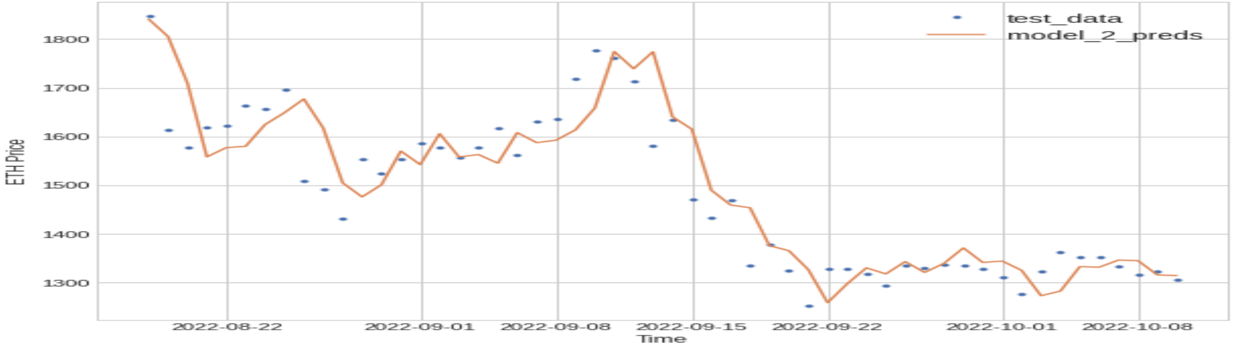


Fig 20: Model-2 Predictions

The model should be created in a similar manner to model_1 but with a different size of input data. Test the best version of model_3 by loading in the best version and evaluating it. In this case, the predictions will be made at a rate of 7 steps at a time (this is the size of the HORIZON). Model_3 results need to be calculated - these will be multidimensional as a result of the data being multidimensional. There is more than one step in the process of predicting, so we are trying to do it all at once.

This will allow you to plot model_3_preds by aggregating them (note that this will reduce the amount of information so the preds will look further ahead than the test data).

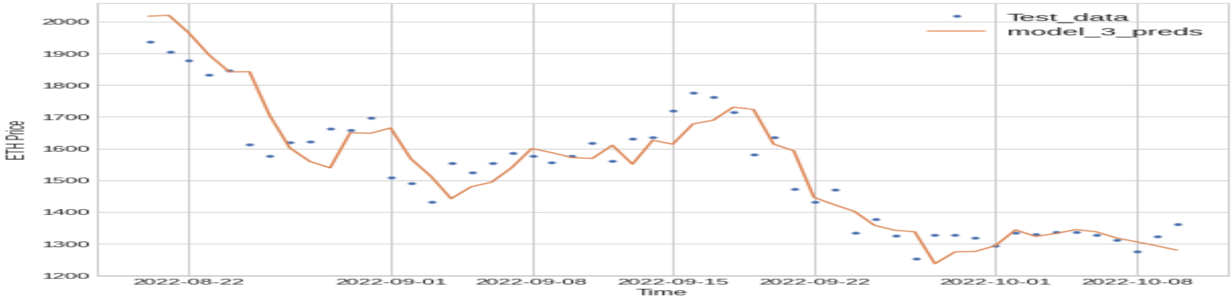


Fig 21: Model-3 Predictions

Create a windowed dataset based on the input data and then set up splits for training and testing. We make sure the shapes of the data samples are correct. In order to make sure that our data is shaped correctly before we pass it on to the Conv1D layer, we need to reshape it as much as possible and enhance the timesteps by adding a new dimension. The first step is to create a model, compile it, and fit it. Conv1D model should be loaded in and analyzed on the test data in order to determine its performance. Predictions are made and the results are evaluated based on the predictions.

A Model using LSTM (Long Short Term Memory):

Since the LSTM is a class of a recurrent neural network, before we can proceed with LSTM, it is imperative that we first understand neural networks and recurrent neural networks in general.

Neural Networks:

Artificial neural networks are layered structures of connected neurons, inspired by biological neural networks. They are not based on a single algorithm, but on a combination of a variety of algorithms which can be combined together to perform complex operations on a set of data.

Recurrent Neural Networks:

An RNN is a neural network class that can be used to deal with temporal data. Each neuron of a RNN has its own state/memory, so input is analyzed in a manner that is based on this internal state. This is accomplished with the help of loops within the neural network. RNNs possess recurring modules of ‘tanh’ layers in order to retain information. This information, however, is not retained for a very long period of time, which is why we need LSTM models to replace them.

LSTM (Long Short Term Memory):

A recurrent neural network is a special type of neural network that can be used for learning long-term dependency relationships in data. This is achieved through the recurring module of the model being created with a combination of four layers that interact with each other at different points in time.

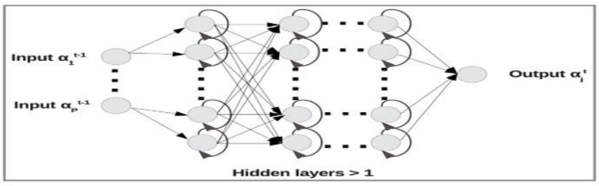


Fig 22: Recurrent neural network

As we proceed, we will make predictions based on our LSTM model. Visualize the block reward/price over time using a graph. It is important to note that because of the differences between the scales of each value, we will scale them so they are between 0 and 1. A minmax_scale variable should be imported from sklearn.preprocessing. Using MinMaxScaler, the data set is rescaled so that all feature values are in the range [0, 1]

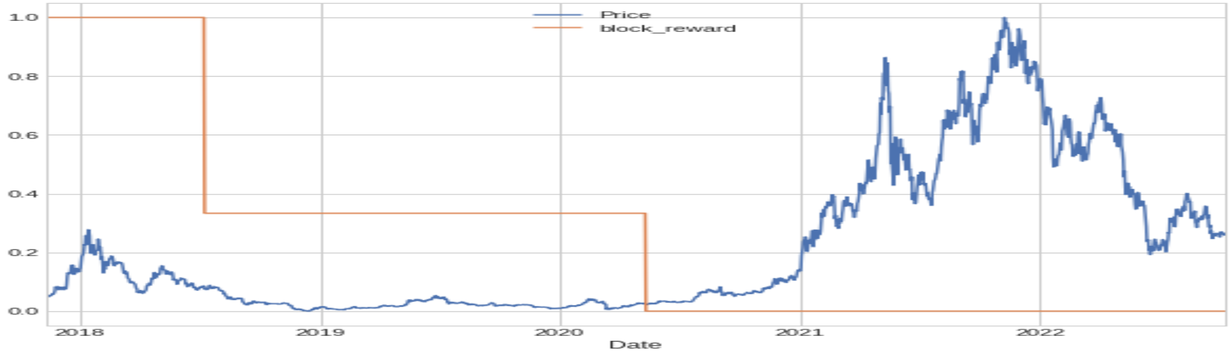
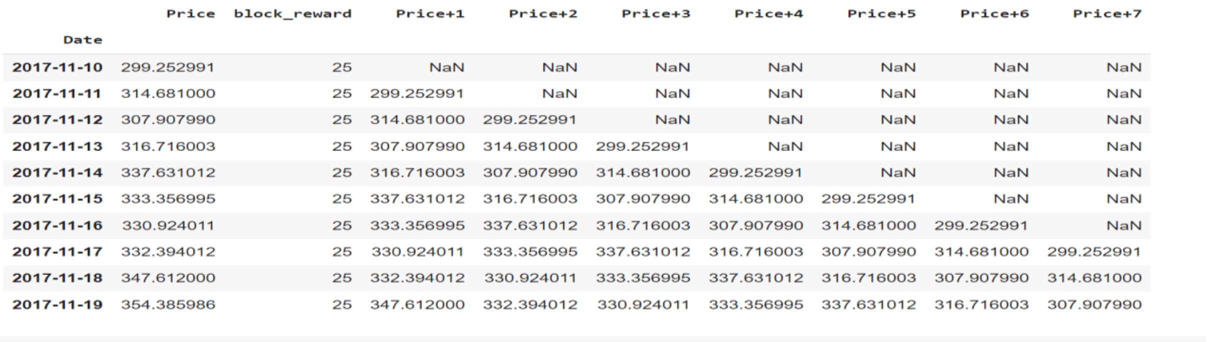


Fig 23: Plotting Ethereum block-reward price over time

Let us now create hyperparameters for the dataset:

It's important to note that hyperparameters are variables that affect the learning process by defining the value of the parameters in a model that a learning algorithm eventually learns.

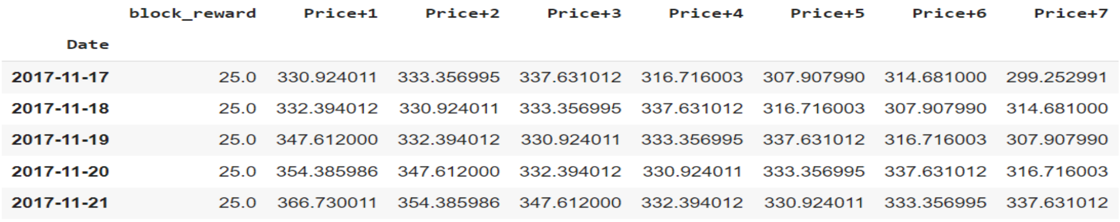
Make a copy of the Ethereum historical data with the block reward feature in order to analyze it. Add windowed columns to WINDOW_SIZE and shift values in each step for each step of the window.



	Price	block_reward	Price+1	Price+2	Price+3	Price+4	Price+5	Price+6	Price+7
Date									
2017-11-10	299.252991	25	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2017-11-11	314.681000	25	299.252991	NaN	NaN	NaN	NaN	NaN	NaN
2017-11-12	307.907990	25	314.681000	299.252991	NaN	NaN	NaN	NaN	NaN
2017-11-13	316.716003	25	307.907990	314.681000	299.252991	NaN	NaN	NaN	NaN
2017-11-14	337.631012	25	316.716003	307.907990	314.681000	299.252991	NaN	NaN	NaN
2017-11-15	333.356995	25	337.631012	316.716003	307.907990	314.681000	299.252991	NaN	NaN
2017-11-16	330.924011	25	333.356995	337.631012	316.716003	307.907990	314.681000	299.252991	NaN
2017-11-17	332.394012	25	330.924011	333.356995	337.631012	316.716003	307.907990	314.681000	299.252991
2017-11-18	347.612000	25	332.394012	330.924011	333.356995	337.631012	316.716003	307.907990	314.681000
2017-11-19	354.385986	25	347.612000	332.394012	330.924011	333.356995	337.631012	316.716003	307.907990

Fig 24: Generated windowed first dataset

It has now been resolved that I need to create X & Y so that I can remove the NaNs and convert to float32 in order to prevent TensorFlow errors.



	block_reward	Price+1	Price+2	Price+3	Price+4	Price+5	Price+6	Price+7
Date								
2017-11-17	25.0	330.924011	333.356995	337.631012	316.716003	307.907990	314.681000	299.252991
2017-11-18	25.0	332.394012	330.924011	333.356995	337.631012	316.716003	307.907990	314.681000
2017-11-19	25.0	347.612000	332.394012	330.924011	333.356995	337.631012	316.716003	307.907990
2017-11-20	25.0	354.385986	347.612000	332.394012	330.924011	333.356995	337.631012	316.716003
2017-11-21	25.0	366.730011	354.385986	347.612000	332.394012	330.924011	333.356995	337.631012

Fig 25: Generated windowed second dataset

Creating a multivariate time series model using time series data. I'm going to add about an extra layer here so that we can beat the naive model by using layers. Dense (128, activation="relu"), I would like to create a function which would use a list of trained models to make and return a list of predictions as well as make predictions based on the current model. Make NBEATSBlock layer dummy by passing dummy inputs to it and we get the forecast: -0.7554213404655457.

The full historical data can be used to train a model to make predictions about the future based on the past.

It is necessary to train the model on the entire data in order to be able to predict the next day's outcome. So, we create and compile the model is the future forecast for the next 60 days.

Now, let us plot a graph to see the future price predictions of Ethereum starting from 2022/10/01 to next 60 days.

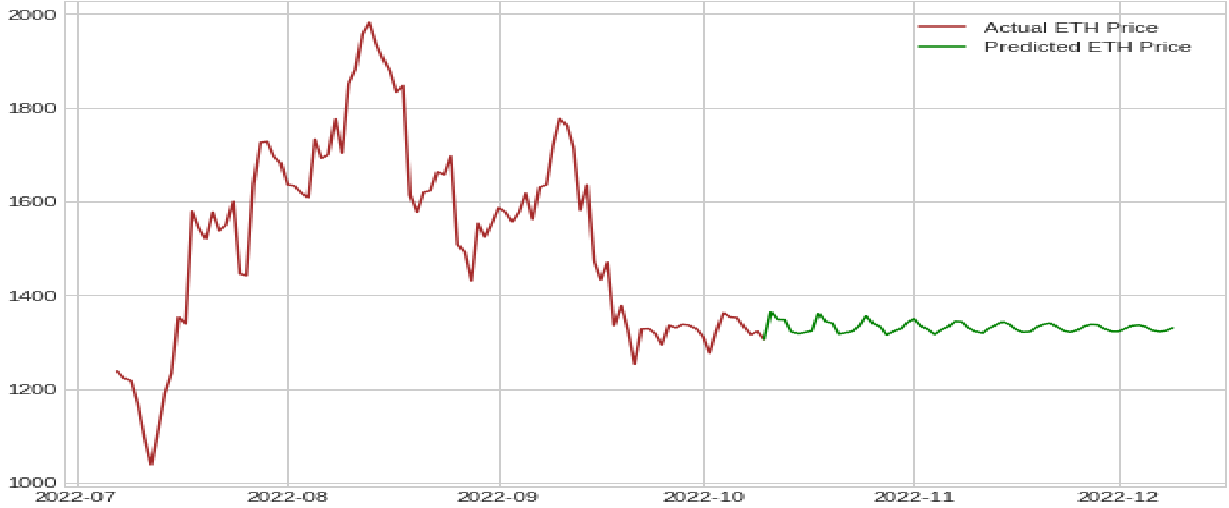


Fig 26: Predicted Price of Ethereum

Discussion:

I've observed something we've seen before, which is that crypto, specifically Bitcoin and Ethereum, are risky assets," "They go up and down just like stocks." Stocks sold off yesterday and today, and crypto pretty much did the same thing." Cryptocurrencies are extremely volatile, and the Russian attack on Ukraine has made them even more volatile than usual. As you can see, the crypto market is “somewhat normal” right now since it's been down for a while. The Russian invasion of Ukraine has brought cryptocurrency into the spotlight.

After reaching a high of more than $68,000 in November 2021, bitcoin is now hovering around $20,000. It's been a tough year for the crypto community. The short-term temperaments are being met with both positive and negative factors that guide the crypto community from here now with bitcoin coming off a nearly two-year low. In addition to institutional investors, venture capital activity is another promising sign for crypto.

Conclusion:

In order to make more accurate predictions about the price of Ethereum, this project makes use of machine learning to forecast the price of these digital currencies. By using historical Ethereum prices, we trained nine models for Ethereum cryptocurrency price prediction. The system is designed to forecast cryptocurrency prices by using LSTM models. The data used is from the past five years (2017 to 2022). I believe this system is adequate for aiding in price prediction, and the results obtained from predicting prices using machine learning appear to be accurate in any situation where there is a technical trade indication, thereby lowering its accuracy during price prediction evaluation. While Ethereum's competition and other factors continue to make it volatile, there's a general feeling of optimism that it'll survive. The NFT (Non-fungible tokens) market is dominated by Ethereum with over 90%. Ethereum's next few months are going to be crucial. It was nearly a year ago that it was at $4,800, but now it's back below $2,000. Whatever happens, investors need to see improved demand and functionality for ether's price to continue rising. There is a good chance that bitcoin transactions can replace traditional currency in countries with technology. However, in developing countries without technology, this would prove to be a disaster. This is the prediction of the actual Ethereum based cryptocurrency price in comparison with the predicted price by the neural network model, and the prediction of a price hike.

APPENDIX-A

Legal:

A fiat currency is one backed by the government or monetary authority, which has the authority to issue it, such as the dollar bill which can be traced to the Federal Reserve Board. There is no public or private entity backing cryptocurrency, so it has been difficult to make a case for the legal status of cryptocurrencies in various financial jurisdictions around the world. The legal status of cryptocurrencies has implications for the way they are traded and used in everyday transactions. It doesn't help matters that cryptocurrencies are largely operated outside most existing financial infrastructure. It was recommended by the Financial Action Task Force (FATF) in June 2019 that wire transfers of cryptocurrencies should be subject to the existing requirements of its Travel Rule, which requires compliance with anti-money laundering standards.

In December 2021, El Salvador became the first and only country in the world to accept Bitcoin as legal tender for monetary transactions by allowing Bitcoin to be used as a medium of exchange. Other countries have varying regulations regarding the use of cryptocurrencies. According to Japan's Payment Services Act, Bitcoin is considered to be legal property. China has banned cryptocurrency exchanges and mining within its borders, and cryptocurrency exchanges operating in the country are obligated to collect information about their customers as well as data on their wire transfers. The Indian government is rumored to be drafting a cryptocurrency framework in December of last year.

European Union regulations make it legal to use cryptocurrencies for trading purposes. Regulatory requirements have been imposed by the European Commission in June 2021, establishing safeguards for regulation of financial services that use cryptocurrencies, as well as establishing rules for companies or vendors that provide these services. Derivatives and other products that use cryptocurrencies must qualify as financial instruments.

In the United States, which is the biggest and most sophisticated financial market in the world, there is the Chicago Mercantile Exchange that offers crypto derivatives like Bitcoin futures. It is said that the Securities and Exchange Commission (SEC) has taken the stance that Bitcoin and Ethereum are not securities in the past; however, Gary Gensler stated in September 2022 that cryptocurrency is a security and should be regulated. In light of this stance, it is likely that cryptocurrency's legal status will soon be regulated.[3]

Bitcoin:

Among the many cryptocurrencies that have been created over the years, Bitcoin is considered the first cryptocurrency, and all other cryptos are called "altcoins" (a combination word derived from "alternative coin"). Because of their scalability, privacy, and the scope of functionality that they provide, Bitcoin and some of the biggest altcoins available are some of the best options out there, but it is difficult to say which cryptos are the best ones. There are numerous blockchains that enable Bitcoin's payments and transactions over the internet. It is considered the world's first decentralized cryptocurrency, using blockchain technology to facilitate digital transactions and payments. Rather than using central banks (such as the Federal Reserve in tandem with the U.S. Treasury) to control the money supply in an economy, Bitcoin blockchain serves as a public ledger of all transactions in the history of Bitcoin. An economy's money supply is controlled by a third party (such as your local bank, credit card issuer, and merchant's bank) that verifies transactions. In order to prevent fraud and other unauthorized tampering with the currency, the ledger is also an important tool for proving that a party owns the Bitcoin they're trying to use. Additionally, the use of a decentralized currency can also lead to faster and cheaper peer-to-peer transfers (such as between two individuals from two different countries) than traditional currency exchanges using third party institutions.[18]

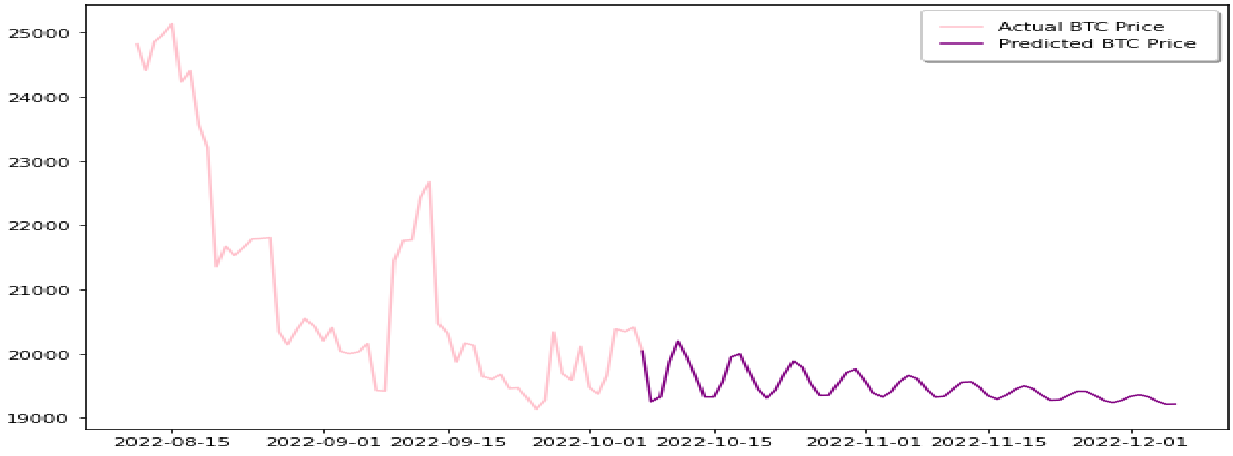


Fig 27: Predicted Price of Bitcoin

Payment:

In order to be able to use your Bitcoins, you should have a cryptocurrency wallet. It is a wallet that contains the private keys to the bitcoins that you own and need to enter each time you are conducting a transaction. Many merchants, retailers, and stores accept Bitcoin as a method of payment for goods and services.

There are many brick-and-mortar stores that accept cryptocurrencies, usually displaying signs that read, “Bitcoin Accepted Here”; transactions can be handled with the proper hardware terminal or wallet address by scanning QR codes or using touchscreen apps, respectively. A business can easily accept Bitcoin payments if it adds this payment option to its other online payment options, such as credit cards, PayPal, etc.[19]

References

1. Ashford, K. (2022, October 11). *What is cryptocurrency?* Forbes. Retrieved November 3, 2022, from https://www.forbes.com/advisor/investing/cryptocurrency/what-is-cryptocurrency/
2. Cointelegraph. (2021, October 29). *What is cryptocurrency? A beginner's Guide to Digital Currency*. Cointelegraph. Retrieved October 23, 2022, from https://cointelegraph.com/blockchain-for-beginners/what-is-a-cryptocurrency-a-beginners-guide-to-digital-money
3. Frankenfield, J. (2022, October 21). *Cryptocurrency explained with pros and cons for investment*. Investopedia. Retrieved October 23, 2022, from https://www.investopedia.com/terms/c/cryptocurrency.asp
4. Hashim1, A. S., Awadh1, W. A., & Hamoud2, A. K. (2020, November 1). *IOPscience*. IOP Conference Series: Materials Science and Engineering. Retrieved October 23, 2022, from https://iopscience.iop.org/article/10.1088/1757-899X/928/3/032019
5. *Ethereum (ETH) price, charts, and news - coinbase*. (n.d.). Retrieved October 31, 2022, from https://www.coinbase.com/price/ethereum
6. By: IBM Cloud Education. (n.d.). *What is Data Science*. IBM. Retrieved October 23, 2022, from https://www.ibm.com/cloud/learn/data-science-introduction
7. *Book draft: Introduction to machine learning*. KDnuggets. (n.d.). Retrieved October 23, 2022, from https://www.kdnuggets.com/2012/11/book-draft-intro-to-machine-learning-alex-smola.html
8. *What is machine learning?* Microsoft Azure. (n.d.). Retrieved October 23, 2022, from https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-machine-learning-platform/
9. By: IBM Cloud Education. (n.d.). *What is machine learning?* IBM. Retrieved October 23, 2022, from https://www.ibm.com/cloud/learn/machine-learning
10. *Supervised vs. unsupervised learning: What's the difference?* IBM. (n.d.). Retrieved October 23, 2022, from https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning
11. Priyadharshini. (2022, September 2). *What is machine learning and types of machine learning [updated]*. Simplilearn.com. Retrieved October 23, 2022, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-machine-learning
12. *11 most popular machine learning software tools in 2022*. Software Testing Help. (2022, September 24). Retrieved October 23, 2022, from https://www.softwaretestinghelp.com/machine-learning-tools/
13. *What is deep learning and how does it work?* Built In. (n.d.). Retrieved October 23, 2022, from https://builtin.com/machine-learning/what-is-deep-learning
14. *Do you know what deep learning is?* Oracle. (n.d.). Retrieved November 3, 2022, from https://www.oracle.com/artificial-intelligence/machine-learning/what-is-deep-learning/
15. *What is deep learning?: How it works, techniques & applications*. How It Works, Techniques & Applications - MATLAB & Simulink. (n.d.). Retrieved October 23, 2022, from https://www.mathworks.com/discovery/deep-learning.html
16. *How (not) to use machine learning for time series forecasting: Avoiding* ... (n.d.). Retrieved November 3, 2022, from https://towardsdatascience.com/how-not-to-

use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls-19f9d7adf424

17. Tyagi, N. (n.d.). *Introduction to time series analysis in Machine Learning*. Analytics Steps. Retrieved November 3, 2022, from https://www.analyticssteps.com/blogs/introduction-time-series-analysis-time-series-forecasting-machine-learning-methods-models
18. Rossolillo, N. (n.d.). *What are all the different types of cryptocurrency?* The Motley Fool. Retrieved October 23, 2022, from https://www.fool.com/investing/stock-market/market-sectors/financials/cryptocurrency-stocks/types-of-cryptocurrencies/
19. Frankenfield, J. (2022, October 21). *What is bitcoin? how to mine, buy, and use it*. Investopedia. Retrieved October 23, 2022, from https://www.investopedia.com/terms/b/bitcoin.asp