UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Deep learning predictive models for non-fungible tokens

Author: **Baudouin MARTELÉE**
Supervisor: **Christian HAFNER**
Reader: **Eugène PIRCALABELU**
Academic year 2021–2022
Master [120] in data science

Contents

Introduction.

A non fungible token (NFT) is a digital identification of a certain asset. This new technology derived from the cryptocurrencies begins in the early 2017. It gives access to functionalities such as identification of digital art piece, valuation of digital collectibles, the avoid the copy-paste principal without giving credits, etc. This market became very influent in the last years. For instance, the number of dollars generated by the NFT market in 2021 amounts to 24.9 billion US dollars. This thesis will deeply analyze the market for different NFTs types and demystify all the growing interest about this new technology that can look disturbing at first sight. The main purpose is to build some deep learning models to predict the market from internal and external views.

This paper is divided in four distinct chapters :

1. The first chapter describes the state of the art in the field of NFT market prediction. This section gives a preview of all the work previously done by the academics.

2. The second chapter will give a detailed definition of an NFT, how it is created, on which technology it relies. This will imply resources that have to be introduced, such as Ethereum blockchain, Smart contract, ERC-20 and ERC-721 protocols. An NFT can be classified in different classes called types. Each of them will be explained in detail in order to produce analyses and see the evolution of the market for every type.

3. The third chapter speaks about the data collection and the analysis. For each type of NFT, a descriptive analysis will be carried out. Furthermore, a global analysis of the market will be visualized.

4. The last chapter will present the elaboration of multiple predictive models in order to have an idea of the future of the NFT market. These models will be computed in different ways : a prediction of the overall market, a prediction of each collection individually and a prediction to a global model taking into account all the collections and the sentiments of the social media Twitter. The impact of social media on the NFT market will be explained in details.

State of the art

First appeared in 2017, the concept of NFT is a very recent technology which implies that the number of academic research papers about it is very limited. This section will describe the current state of the art of NFT's by means of the most relevant papers for this thesis.

The paper "Mapping the NFT revolution: market trends, trade networks, and visual features" produced by Matthieu Nadini and Baronchelli [2021] is one of the only papers really explaining the sales prediction by NFT categories. This paper provides an understanding of the different types of NFTs and an initial comprehensive analysis of NFT's through April 27, 2021. This analysis allows to visualize the evolution of the market as a whole but also by types of NFT. This paper comes to the conclusion that the types of NFT's have different proportions in the global market. Art is the dominant category followed by collectibles, games and the metaverse. All these categories will be explained in this thesis. This paper focused on regression techniques to predict the sales price, constitutes a good start to understand the world of NFT's and conducts a detailed analysis.

The paper of Nicola Borri and Tsyvinski [2022], "The economics of Non-Fungible tokens" talks about the returns of this market. This paper focuses more on a financial analysis of the market, taking into account the returns and the volatility of asset prices. This paper relies on time series and regression with variables like volatility, valuation ratio, attention, past returns and volume.

"Non-Fungible Tokens (NFT). The Analysis of Risk and Return", a paper written by Mazur [2021], also examines the risk and return of the NFT market with a comparison to the leading cryptocurrency Bitcoin. This paper, however, makes many basic assumptions about the market's performance which makes the analysis very specific.

As said before, recent papers released in 2021 or early 2022 do not exploit deep learning methods in order to analyze the NFT's market. One of the main purposes is to predict the market with the help of advanced deep learning techniques.

Chapter 1

About non-fungible tokens

1.1 Non-fungible token definition

The definition of an NFT can vary very slightly from one research paper to another as observed in the papers of Chohan [2021] and Ante [2021]. However, an NFT can be defined in one sentence:

"**A non-fungible token is a digital asset that represents a line on a blockchain.**"

This sentence may seem unintelligible; therefore, we will define the central words to really understand what an NFT is.

Let's begin with the meaning of " non-fungible ". An non-fungible object is a unique and non-interchangeable object. In the case of NFT's, two objects are not interchangeable because they do not hold the same value. There is nothing better than providing an example to explain this concept of "non-fungible": For instance, a bitcoin is fungible because two bitcoins hold the same value and then we can exchange them without loss or gain for either exchanger. In contrast, if we consider two items of an art collection, they can have different values. With this concept comes the notion of scarcity of an object. The rarer an object is, the higher its value.

The term "digital asset" refers to an ownership with any kind of data in binary form stored in your computer or on the internet. Toygar et al. [2013]. This data can consist of an image, a text, a video, etc.

The term "Blockchain" will be explained further in detail. For now, imagine the blockchain as a book that records objects on it. Every object is identified by a line on the book with his associated value.

After explaining all the main terms of this sentence, it could be summarized by saying that an NFT is a system of unique identification of an digital object stored in a virtual book. This identification brings scarcity and value to the object. But what exactly can these objects consist of ? They may include diversity and be used for different purposes like art, collectible, domain, music, in-game items, etc. With this notion in mind, some questions may arise in our minds. Knowing that it is possible to replicate any digital object easily on the internet, why would one buy NFT? Which rights are held by buying an NFT? If an artist puts a digital art online, it is possible for everyone to see this digital piece and easily replicate it by copying and pasting it.

How could we then give value and scarcity to a replicable object? The solution could be to put an NFT on this piece of art and then to register this piece on the blockchain. This registration will assure the presence of an identification token that makes it unique. The scarcity comes with the number of NFT's that exist on the market.

At first sight, when you look at the two figures below, there is no significant difference between them. All pixels of both figures are exactly the same and hold the same RGB value, in other words, nothing differs. However, one is worth nothing while the other holds a value of 360k dollars. The first one is just a screenshot of the Apes Bored Club 300, found on Google Image. The second one is the exact same image hosted on a NFT market and registered on a certain blockchain. Even if the second example is a fictive one, because the thesis writer doesn't actually hold such an NFT, the example here shows that distinguishing an NFT from another image is impossible.

Figure 1.1.1: Picture of the BAYC 8108 found on Google. Source

Figure 1.1.2: NFT of the BAYC 8108 hosted on the blockchain

Now that the concept of NFT is no longer a secret, a question is raised: where does one buy a NFT ? Some marketplaces take care of everything.

1.2 NFTs markets

Several marketplaces have opened since the beginning of the NFT revolution. A marketplace could be defined as an online shop for NFT's, it may rely on different blockchains like Ethereum, Polygon or Solana. The biggest marketplace for NFT's is called Opensea. Opensea takes place in Ethereum and Polygon blockchains. Data will be collected and analyzed from this marketplace. All the data that has been collected for the writinf of this thesis rely on the Ethereum blockchain. These markets have developed a way to collect a file, hash it and then store the hash on the blockchain. Naturally, with every hash comes the corresponding price of the NFT. These platforms are available both for sale and purchase. Here stands a list of the main marketplaces :

1. Opensea
2. Rarible
3. Superare
4. Axie Marketplace
5. Larva Labs/CryptoPunks

It is interesting to notice that some marketplaces, like Axie Marketplace or Larva Labs/CryptoPunks, sell only one typical collection.

1.3 Technical part

1.3.1 Blockchain

In 2008, an unprecedented financial crisis plunged the world into a period of chaos where people's trust in banks felt drastically. Banks are centralized, meaning that money is managed by a third party. Towards the end of 2008, a decentralized technology appeared. Created by an anonymous person (or a group of people) named Satoshi Nakamoto, this technology would revolutionize the digital world in different aspects, its most famous application being the use of cryptocurrencies.

To understand this technology, let's first introduce the basic functioning of a bank. Banks regulate the transaction between users by a third party. This tier can be disturbing in the sense that people entrust their money to someone else. The bank, when someone entrusts it with his money, can do whatever it wants with it . The idea behind blockchains is to be able to free oneself from this third party. Figure 1.3.1, shows the fundamental difference between centralized and decentralized financing. Moreover, the blockchain guarantees anonymity behind each user of it.

The blockchain, hence its name, is a chain of blocks where each block is in fact a page where transactions are recorded on, cf. Shahar Somin and Altshuler [2020]. Multiple

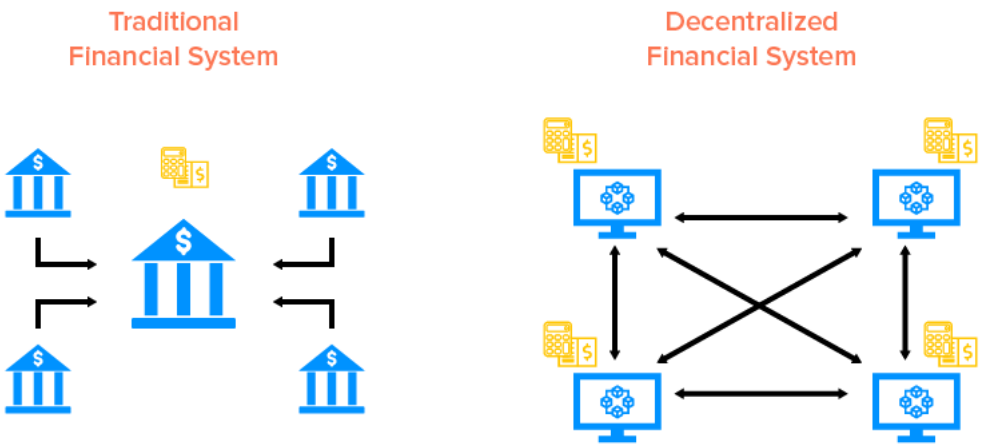

Figure 1.3.1: CeFi vs Defi, Source

blockchains exist, like the Bitcoin blockchain, the Ethereum blockchain,etc. However, regardless of the blockchain, the default transaction is composed of at least a sending address, a destination address and an amount. Moreover, a block is also composed by two headers that give information about the block Belotti et al. [2019]. The users are identified on the blockchain by addresses. These addresses are anonymous because they are represented in the hexadecimal form of "xFF256418769656". Both private and public blockchains exist, but NFT's only focus on public blockchains. In this kind of blockchain, each page of the register can be consulted by anyone. Each time a transaction is made by a user, another user called the "miner" will write this transaction on a page. A page contains a limited amount of transactions, which implies that as soon as this page is complete, a new page is created in the register. One could think that, in this process, the third party would be the person who registers the transaction on a page (the "miner") and who will take a commission and manage the register of accounts, but it is not. In fact, blockchains are based on a principle of peer-to-peer system, which means that nobody manages the blockchain, and the "miners" are the ones who create the whole ecosystem. The miners generate new blocks in the chain and receive a reward in return. They are also the ones who validate each transaction of the blockchain. The transactions are validated unanimously in relation to the computing power used by the miners; this principle is called the "Proof of Work". This concept will be explained in more detail in the next section. The blockchain is impossible to hack unless you control 98% of all the computational power of all the users combined. This makes it an almost infallible and very popular system.

The blockchain is often characterized by three words :

- Security
- Decentralization
- Anonymization

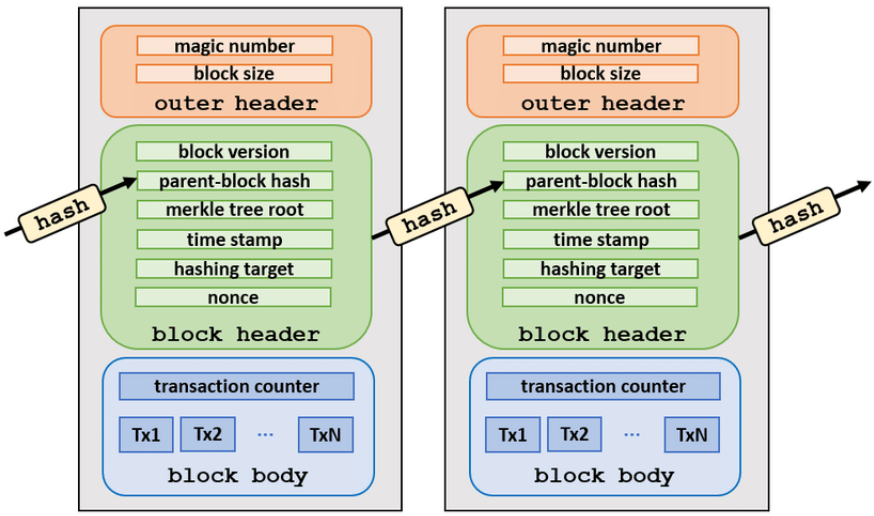

Figure 1.3.2: Blockchain representation. Source: Belotti et al. [2019]

1.3.2 Proof of Work protocol

The "Proof of work" protocol involves asking miners to solve a complex mathematical problem that requires a lot of computing power. The first person who is able to solve the problem will also be the next one to create blocks on the blockchain. Therefore, miners apply a hashing algorithm to the same set of data until they find the expected result.Porat et al. [2017]

In the Bitcoin protocol, miners perform two consecutive SHA256 hashes (it is the most secure hash) on block headers. Each new hash value depends on the block value and a nonce, which is a random number chosen by the mining software. For a block to be valid, this hash must be less than the mining difficulty.

This difficulty of calculation increases with time, more precisely all 2016 blocks. It takes powerful graphics cards to mine a cryptocurrency.

1.3.3 Ethereum blockchain and smart contracts

The first blockchain having been created is the bitcoin blockchain and it is only in year 2015 that another blockchain with new specificities appears. This blockchain, that will later give birth to the concept of NFT, is known as Ethereum blockchain, whose creator is Vitalik Buterin. The most interesting feature of this blockchain is the possibility of creating smart contracts. Until now, the basic blockchain allowed one to make simple transactions,but programming a secure transaction was not yet possible, but this is where the role of smart contracts come into the picture. Let's look at an example, if user A asks user B to perform a service for 10 Ether (the cryptocurrency of the Ethereum blockchain) but that user A

doesn't want to pay until user B has completed the service, that would create mistrust: how can user B be sure that user A will pay the requested amount? The solution could be to create a smart contract, consisting of user A needing to pay the 10 Ether but these not being transferred until user B fulfills the task in time. It is now possible to create all kinds of tokens that satisfy certain rules established by the smart contract. Smart contracts are built according to protocols to follow. It is interesting to describe the two most-used protocols: ERC-20 and ERC-721.

1.3.4 ERC-20 & ERC-721 protocols

Before introducing the ERC-721 and having a better understanding of it, an overview needs to be done about the ERC-20 Protocol. The ERC-20 protocol is a standard protocol used in a smart contract on Ethereum blockchain for the creation of tokens.Cuffe [2018]

ERC-20, appeared in late 2016, establishes that every token created from the same smart contract has the same value and that secure transactions are available.

To make a smart contract following with ERC-20, six mandatory functions must be implemented. Khalid Husain Ansari [2020]

- **totalSupply()** : There is a limited amount of tokens produced by the smart contract. When this number is reached, it stops the production of tokens.
- **balanceOf(address user)** : Returns the number of tokens that are held by a certain address on the blockchain.
- **transfer(address receiver, uint256 amount)** : When creating a smart contract, a user needs to be specified for the first time to hold all the tokens created. This user is called the "owner of the contract". This function allows this owner to transfer an amount of tokens to another address.
- **transferFrom(address sender, address receiver, uint256 amount)** : Allows a user to transfer an amount of tokens to another user.
- **approve(address spender, uint256 amount)** : The smart contract verifies if it can transfer tokens to a user by checking if the total supply is reached or not.
- **allowance(address owner, address spender)** : The smart contract verifies that the user wanting to transfer tokens has the minimum amount needed for the transfer.

ERC-20 gives the possibility to be more precise about the creation of a certain token. Three other optional functions exist :

- **name()** : gives a name to the token
- **symbol()** : gives an abbreviation for the token. For instance : BTC stands for Bitcoin, ETH for Ethereum, ...
- **decimals()** : the number of decimals the token uses (8 by default). A decimal of 8 means dividing the token amount by 100000000 to get its user representation.

It is important to note that this protocol is not valid for the creation of NFT, for multiple reasons.The first one is the notion of rarity that must be considered. In the ERC-20 protocol, however, there is no function able to define this notion and therefore every token holds the same value. For instance, when you create two ETH, these ETH coins hold the same value. For NFTs, since the value of two different tokens of the same collection can have different values, a bid can be made for a certain NFT and the owner can accept or deny it. We have to represent this notion of allowance by an owner. With this concept of rarity and unique token comes the concept of "Ownership" of a token by a user. The ERC-721 protocol is introduced to resolve these constraints that the ERC-20 can't handle.

ERC-721 represents the first protocol ever done for NFT in 2017. It represents the most popular protocol, even if the ERC-1155 protocol is also used for NFTs.
The ERC-20 and ERC-721 have some functions in common Entriken et al. [2018] :

- **name()**
- **symbol()**
- **totalSupply()**
- **balanceOf()**
- **transfer()**

There are different functions about the ownership of a particular token.

- **ownerOf()** : Returns the address of the owner of a token.
- **approve()** : The owner of an NFT can give the permission to transfer tokens to another user.
- **safeTransferFrom()** : An evolution of the transfer from the ERC-20.
- **takeOwnership()** : (Optionnal) When a user needs to get some tokens from another user account.
- **tokenOfOwnerByIndex()** : (Optionnal) Enumerates the tokens held by an holder.

1.4 NFTs categories

An NFT can represent any object likely to be digitized and it leads to the creation of an almost unlimited number of elements. Humans tend to find a way, just like they would do with any huge amount of data, to sort it in order to identify an NFT and classify it in a family of other assets with similar characteristics. The number of existing categories varies from one literature to another.

However, in line with the site www.nonfungible.com (version of 12 october 2021), at least five main categories of NFT can be identified:

- **Art**
- **Collectible**
- **Game**
- **Metaverse**
- **Utility**

A last category, called "Other", could be created for other NFTs; this category could group sub-classes of less represented NFTs such as music, domain names, collection cards,... . An other class "Defi" designs all the decentralized finance but, as it is not essential in this paper, we will not analyze it in details. We will now describe the subcategories mentioned earlier in more detail.

1.4.1 Art

Art is the category of NFT that is most familiar to the public. It generates the biggest value in term of trading in the market and is the type that is the most profitable. For instance, Beeple, one of the most famous NFT artists, sold one of his creations, "The Verge" for more than 69.3 million dollars (Figure 1.4.3).

In this paper, an analysis about the two biggest collections will be conducted. **CryptoPunks** and **Bored Apes Yacht Club** are everywhere on the web. When the subject of NFT's is discussed, these two collections impose themselves directly on the mind. Figure 1.4.1 and 1.4.2 represents some assets of each collection.

Figure 1.4.1: CryptoPunks. Source.

Figure 1.4.2: Bored Ape Yacht Club. Source.

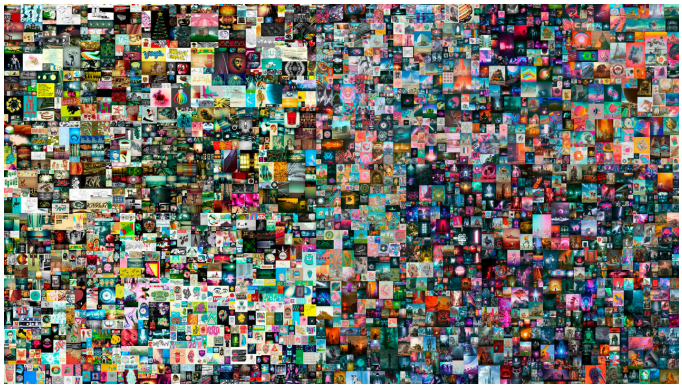

Figure 1.4.3: "The Verge" by Beeple. Source.

1.4.2 Collectibles

Unlike art, the collection represents a set of images with a basic entity then modified by different attributes.These features can add value and rarity to elements of the collection depending on their rarity.

The rise of this category appeared with a collection called Cryptokitties, which had 10k NFTs at its release. This collection contains a variety of colorful cats, drawn in a comic book style. To date, the notoriety of CryptoKitties has decreased due to the creation of new Cryptokitties; the latter made their number increase from 10k to 2 million cats. This growing number has caused the collection to lose value over time. Additionally, two well-known examples of collections will be introduced to better understand this key category.

CryptoPunks represents a collection of 24x24 pixel images stored on the Ethereum blockchain. Cryptopunks are separated into 5 classes: male, female, alien, zombie and monkey. A cryptopunk can contain up to 7 attributes. This constitutes one of the most valuable collections.

Like Cryptopunks, Bored Ape Yacht Club is a collection containing 10,000 staff. They are also hosted on the Ethereum blockchain. One of the main goals of this paper is to analyze the characteristics of each image within both collections in order to visualize whether or not the attributes could be correlated with the price.

This category is one of the most attention-grabbing and money-making. In fact, the rise of well-known collections such as Cryptokitties, CryptoPunks and Bored Apes Yacht Club has led to a growing interest from investors. Sometimes Collections are classified in both categories, namely "Art" and "Collectibles", like Cryptopunks and Bored Ape Yacht Club.

Figure 1.4.4: Collectible "Koala Intelligence Agency". Source.

1.4.3 Game

The three biggest games on the market today are called Axie Infinity and Rumble Kong League.

Axie is a digital pet community centred around collecting, training, raising, and battling fantasy creatures called Axie. Each Axie has unique genetic data stored on the Ethereum blockchain. An Axie can possess 6 out of hundreds of possible body parts. Each body part has its own battle move, meaning that the combinations for creating unique little battlers are infinite! Axie Infinity is the first game built on the blockchain to feature animated characters and rich, immersive gameplay. The game was created in order to empower gamers by giving them true ownership of their game assets as well as to make blockchain technology accessible to anyone who has ever played a game like Pokemon or Tomogochi. Axie Infinity is the biggest NFT game in term of sales, but there is no API available to fetch data and it is impossible to scrap the website due to hundreds of thousands of transactions per day.

Rumble Kong League was launched with a set of unique 10,000 ERC-721 tokens, called Kongs. Each Kong comes with a randomly generated set of attributes that can influence its performance in a match, providing additional depth and strategic elements.

Each Kong is assembled from a set of 100+ carefully hand-drawn traits, split into various categories. Out of over 45 million possible combinations, 10.000 will be assembled through a programmatic approach, giving each Kong a distinctive look. This collection joins the collection to be analyzed for this paper. Rumble Kong League is still in development and the apes will be used as players for basketball games. Nevertheless, the market is already booming.

Figure 1.4.5: Rumble Kong League. Source.

1.4.4 Metaverse

Metaverse, gaining considerable popularity, is considered by some as the future of social interactions or the "real estate" of the web.

The metaverse is complicated to define due to its abstraction but it could be described as a virtual world where social interactions are possible. These interactions are carried out by means of avatars, virtual characters that represent people in this virtual world.

Without a proper corresponding visual representation, the concept of a virtual world is very abstract. It is therefore important to understand that any virtual world can be created and that an infinite number of worlds are possible. An in-depth analysis of two specific worlds will reveal the complexity of this category.

Decentraland is a virtual world where one can buy parcels of virtual "Land". DCTL is limited in space. It is represented by a square divided into plots. The square's dimensions go from -150 to 150 of side for a total of 90000 individual plots. If we took a closer look to a Decentraland map, we could see there are some characteristics to highlight: some roads are located in precise places of the map in order to be able to move in this metaverse. In the figure below, a gameplay demonstrates a concrete example of it.

Decentraland also provides accessories to add to the land, such as houses, walls, animals, clothes for the avatars and an infinite number of other possibilities. It contains an entire marketplace.

Sandbox is also a virtual world that promotes more social interactions than Decentraland. A large community has formed over the last few years, causing the metaverse to be in constant evolution.

These platforms will certainly get recognition in the near future because of the advertisements that are taking a huge place in the society.

Figure 1.4.6: Decentraland gameplay. Source.

1.4.5 Utility

This category refers to a very recent type of NFT that offers to their owners the opportunity to obtain rewards, gifts or services just by holding it. Being very new and constituting a large minority of NFTs, they will not be included in this analysis.

Chapter 2

Data Analysis

2.1 Data fetching

The first important step to take in this research is data collection. This section describes the process of data selection as well as some of the techniques used to collect the data. These data can also be hosted on different platforms which will influence the data collection techniques. This data collection will be described following a timeline, which will be oriented on the theme of this thesis.

What must first of all attract our attention is the current state of the NFT's market. The website https://nonfungible.com/ provides data on this subject. The website contains various data about the NFT market such as the number of daily NFT sales or the number of US dollars generated by these sales, the number of unique wallets and many other features. This information is important to acquire for a global understanding of the market. It is also possible to select these features directly by NFT's category. However, only a few categories are available, such as Art, Collectible, DeFi, Game, Metaverse and Utility. One of the advantages of this site is the possibility to download a csv file directly containing the selected features. Furthermore, data is updated in real time thanks to real time requests sent to the Ethereum Blockchain.

Once the financial data is fetched, the next step is to collect data for each category, such as features. As a reminder, the focus is on four NFT collection : Cryptopunks, Bored Ape Yacht Club, Decentraland and Rumble Kong League. Most of these data was fetched on the popular platform NFT marketplace Opensea.

Opensea offers an API for developers to create decentralized applications. In this case, we are going to exploit this API to retrieve data that will be useful during the exploratory analysis. Opensea makes available data that is accessible to the general public such as statistics of a certain chosen token. However, if we want to access data such as individual assets of a token and their respective attributes, we need an authorization from Opensea

which will then transmit an API key in order to collect this data. This key can be obtained via a form, but the wait is extremely long. To get a key faster, it is possible to contact Opensea directly on the corresponding channel on Discord.

Through this API, it will be possible to issue requests to retrieve the transactions made in real time. There are different types of status for a transaction which can be "successful", "cancelled", "bid", "transfer", etc. During this analysis, only "successful" transactions will be analyzed. Thus, we can quantify the sales and the associated amounts. The collections we are interested in are, initially, the Cryptopunks and the Bored Apes Yacht Club. We will therefore take interesting variables such as the name of the asset, the price at which it was sold in ETH and in USD, the date/time of the transaction as well as the different attributes of this asset.

For Decentraland and Bored Ape Yacht club, the information given by Opensea is very useful but does not include all the interesting information. To absorb as much data as possible, the website www.nonfungible.com will also be scrapped. This technique consists of polling a website against its HTML elements and entering the data into a database. This technique is effective but is restricted depending on the websites that can detect the scrapping and stop your script.

Last but not least, sentiments are a main requirement for this thesis. The social network with the most powerful influence on NFT market is Twitter, one of the most influent social media on Earth.Twitter has an API that allows for the retrieval of Tweets. Tweets can be retrieved through queries. Keywords can be inserted into queries to target tweets, as well as the dates and publishers of those tweets.

The query to the twitter API is thus composed as:

- Keywords : nft, cryptopunks, boredapeyachtclub, bayc, decentraland, rkl, rumblekongleague.
- Dates : from 2021-01-01 to 2022-01-01
- Publishers : the top 20 NFT and crypto influencers of 2021, including Elon Musk, Garry Vee, rac.eth, OhhShiny, VitalikButerin, etc.

Notice: An twitter API key is required to fetch data on Twitter. It takes time to receive it.

2.2 Data transformation

The information collected through the APIs is raw data, in the form of a JSON file. We will need to transform this data into a dataframe so that we can use it for our data analysis. The data from the web scrapping can be directly put into a dataframe form thanks to a python package called Beautiful Soup. However, a transformation of the data using the

Numpy and Pandas packages of python allows to create a usable dataframe gathering all the transactions and the corresponding prices. This data collection and transformation step represents a significant part of the whole thesis because of the time that all the steps take (API key request, API calls, web scraping, dataframe handling, etc).

2.3 Data analysis

2.3.1 Global analysis

An initial global analysis will be performed to understand the 2021 NFT market. We need to determine which category is the most popular and how each category changed over time. To begin, on the first three figures below, an analysis of the number of sales per day or for the entire year will be done.

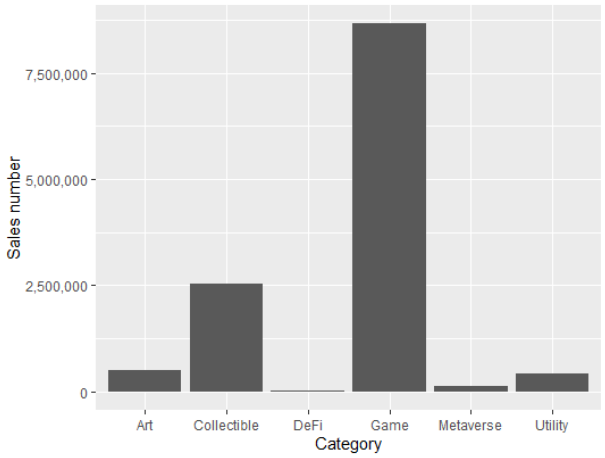

Figure 2.3.1: Sales number for year 2021

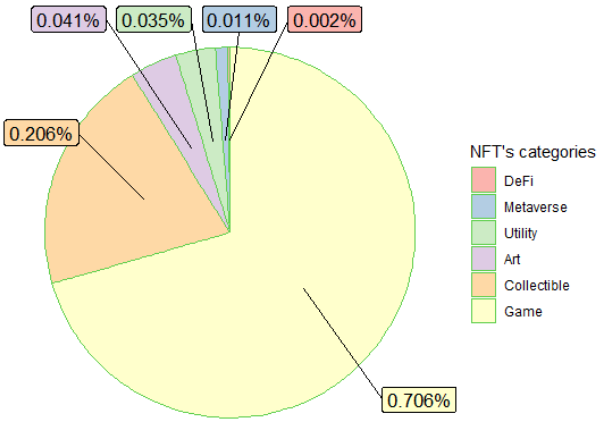

Figure 2.3.2: Percentage of sales of the market for each NFT category

Figure 2.3.1 shows the number of total sales for the year 2021 based on NFT categories. The Game category leads by a wide margin in terms of sales, followed by the collection category, implying that the gaming community appears to be heavily involved in the NFT market. Figure 2.3.2 confirms this by showing that video games sales alone account for nearly 70 percent of sales for the year 2021. This is definitely a category that has seen a meteoric rise in 2021, with the previous year's market largely dominated by art and collectibles.

Figure 2.3.3 shows that 2021 marks the beginning of the NFT's rise in the market. A growing interest is visible. At first glance, we can see that the sales of the games made a huge jump from the summer, around June, to a sharp decline in early October. This is due to one game in particular: Axie Infinity.

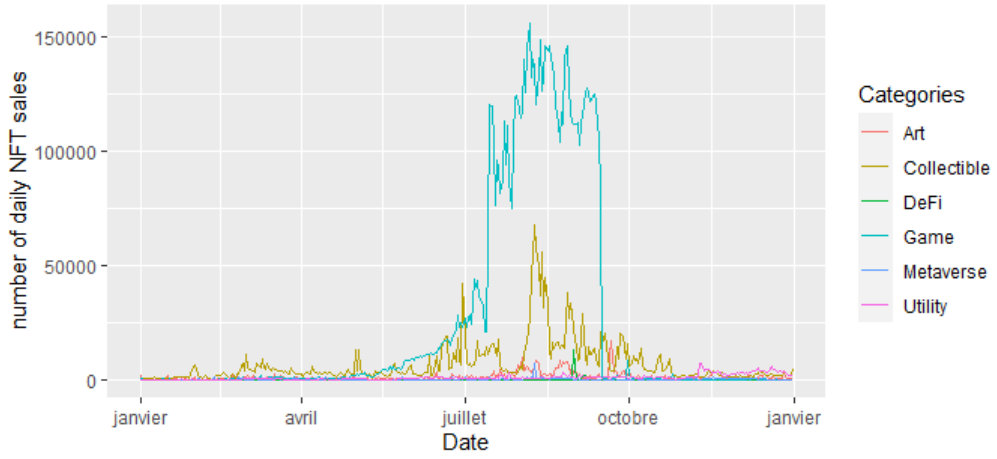

Figure 2.3.3: Number of daily sales for the entire year 2021 by category

The second increase this year is the collection category. The word NFT is now on everyone's lips and is becoming a well known concept and is considered by many to be the future of the art market. One of the aspects that particularly attracts buyers is the fact that this market is speculative. Many can make very large sums of money very easily with minimal knowledge. Others view NFTs as a long term investment by investing in NFTs with very high price floors such as Cryptopunks or Bored Apes Yacht Club.

So far, the number of sales has been analyzed, we now need to focus on the financial aspect to see which category makes the most money. The three figures below analyze the USD prices generated by each NFT category.

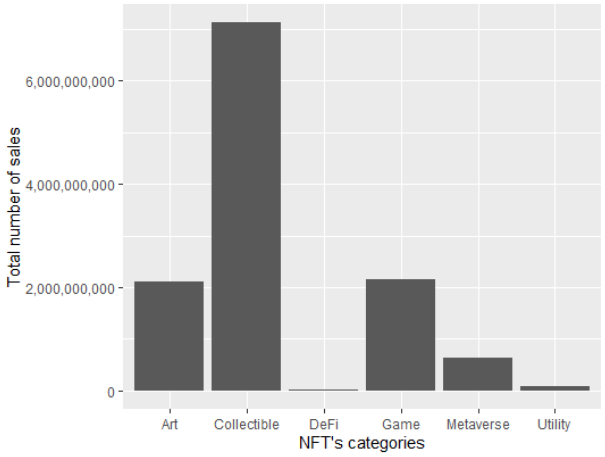

Figure 2.3.4: USD volume for each category

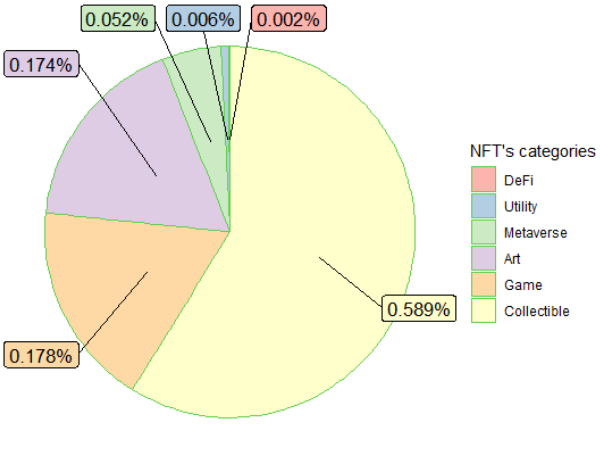

Figure 2.3.5: Percentage of USD volume for each category

There is a real difference between the number of sales and the revenue they generate. The category that produces the most sales does not necessarily generate the most revenue in the market. Figure 2.3.5 confirms this by showing that collections generate the most volume, accounting for nearly 60% of the market, followed by art and games, each accounting for nearly 17%. The figure 2.3.6 shows the daily volume generated for the year 2021 for each category

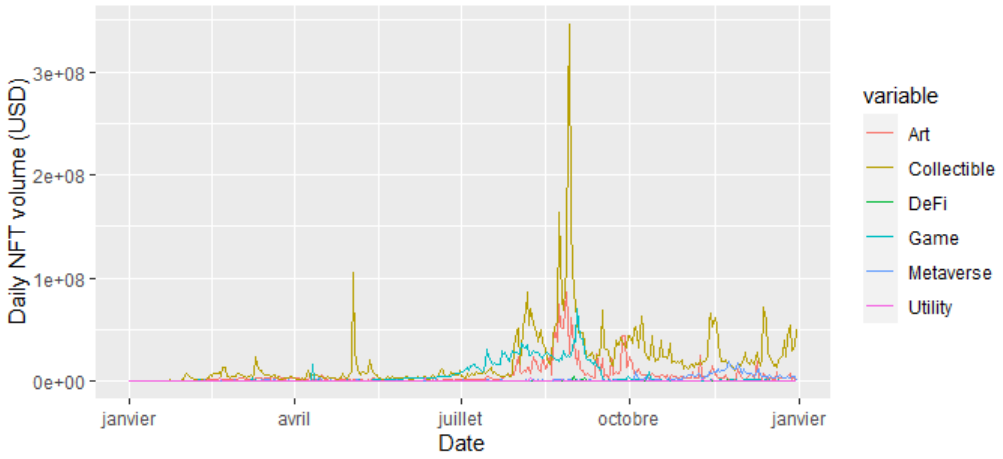

Figure 2.3.6: USD volume for the entire year 2021 by category

After our analysis of the market as a whole, each category will be inspected separately in depth to see the trends in each of them during the year 2021.

2.3.2 Cryptopunks analysis

During the year 2021, 37,800 transactions were recorded on the blockchain. As a reminder, these are the transactions made on the Opensea platform and whose transactions have been classified as "successful". From these 37,800, we remove the transactions whose prices have a zeo value, and, we end with a dataset of 32,104 transactions. A visual representation of transaction prices is essential to understanding the volume that Cryptopunks generate. After adding up the transactions per day, we can visualize the trends over the year 2021 on the following figure 2.3.7. From the end of February, an increase in prices is visible until the end of May. Thereafter, the market drops and stabilizes until the end of July when a boom appears with the month of August when the Cryptopunks became known to the general public. At the end of October, a big peak appears. This is due to the sale of a single NFT (CryptoPunk 9998), worth 533 million US dollars. This sale is the largest recorded sale of all time; however, it is often not accounted for because the buyer of this NFT is also the seller. This could indicate a fraudulent sale.

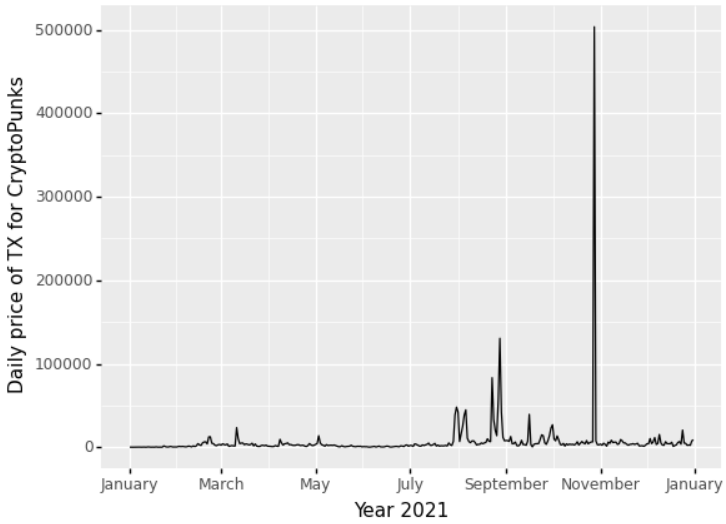

Figure 2.3.7: Daily price of TX (ETH) for CryptoPunks (2021)

The table below gives statistics to get a better overview of the data. Note that the mean is almost double the median, indicating a positive skewed distribution.

Table 2.3.1: Descriptive statistics of the price (ETH) for transaction in 2021

	count	mean	std	min	25%	50%	75%	max
Price	32104.0	75.41	1391.22	4.00e-17	21.97	39.0	85.0	124457.07

After discussing and visualizing the price, attention will be paid to the variables and especially the characteristics of the Cryptopunks. These are classified according to different features such as type, gender, skin tone and number of attributes. Depending on the number of attributes, each Cryptopunk has a set of characteristics. This set can contain different attributes such as hair, beard, eyes, headgear, glasses, etc.

Before analyzing the first data, it is essential to remember that there are a total of 10,000 Cryptopunks. According to the figures on the official website https://www.larvalabs.com/cryptopunks, here is the official composition of the Cryptopunks :

Table 2.3.2: Cryptopunks count depends on their types and genders

Type and Gender	Alien	Ape	Zombie	Human	Male	Female
CP count	9	24	88	9879	3840	6039

Now that the official numbers have been released, we can look at the bar charts below to see the different distributions of the categorical variables for each 2021 CryptoPunks transaction. For the type category, Human is far ahead, which is not surprising given the original distribution of this category in the original 10,000 Cryptopunks. For gender, the percentage of males is 66.6% against 33.3% of females in the transactions, compared to the original distribution of 60% males and 40% females.

Table 2.3.3: Cryptopunks count depends on their attributes number

Attributes count	0	1	2	3	4	5	6	7
CP count	8	333	3560	4501	1420	166	11	1

Table 2.3.4: Cryptopunks count depends on their skin tone

Skin tone	Albino	Dark	Light	Medium	Other
Cryptopunks count	1018	2824	3006	3031	121

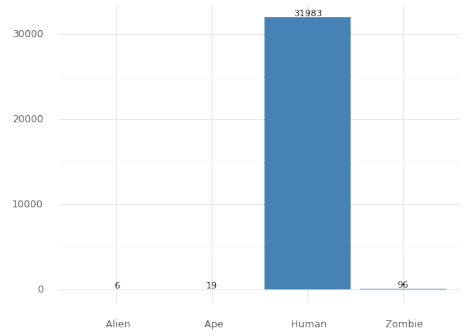

Figure 2.3.8: Count of the type traded

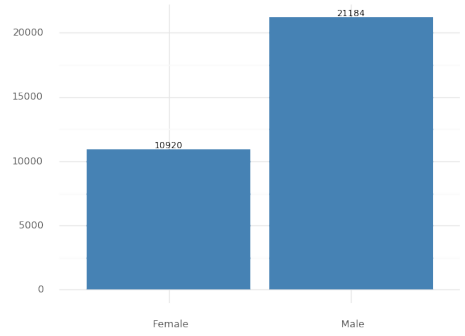

Figure 2.3.9: Count of the gender traded

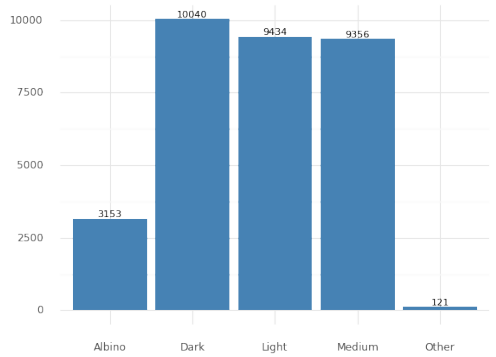

Figure 2.3.10: Count of the skin tone traded

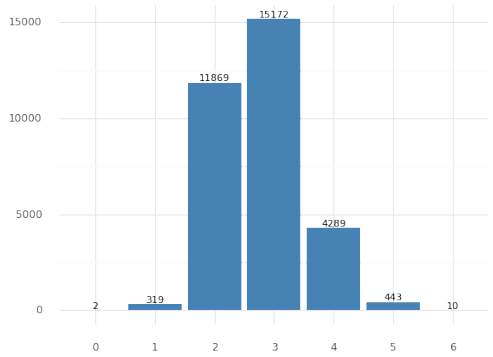

Figure 2.3.11: Count of attributes traded

On Skin tone table, the "Other" category for skin tone represents other categories such as aliens or zombies that have different skin tones such as blue or green. The majority of transactions in the year 2021 are focused on Cryptopunks that have between 2 or 3 attributes. We will see later that increasing number of attributes does not necessarily mean increasing the price. One of the main objectives of this thesis is to understand what intrinsic characteristics of an NFT can influence its price. In other words, can its type, gender, skin and attributes such as earrings, headgear, etc. influence its price. Figure 2.3.12 shows the top 8 characteristics present in 2021. This is an overview of the intrinsic trend of Cryptopunks. Many of the Cryptopunks sold show the presence of earrings, cigarettes, bandanas, etc. Conversely, taking Figure 2.3.13, an asset will sell less if it features the following: Cap, Front Beard, Hot Lipstick, etc.

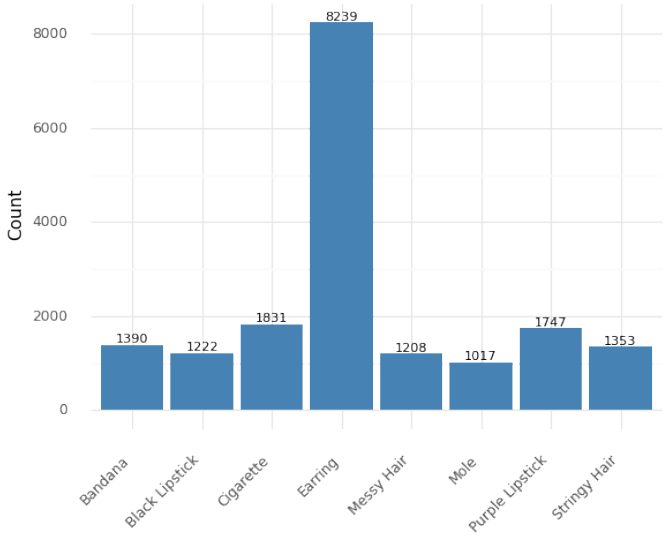

Figure 2.3.12: 8 Biggest traits traded for Cryptopunks during 2021

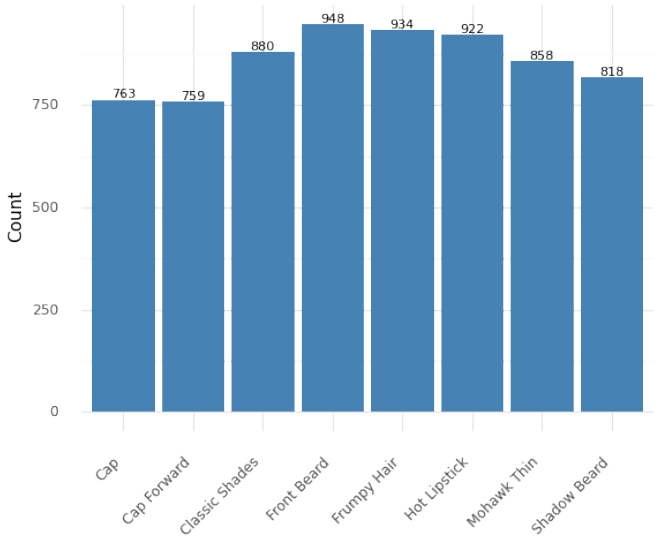

Figure 2.3.13: 8 lowest traits traded for Cryptopunks during 2021

It would also be interesting to analyze which Cryptopunks were traded most often this year and generate sales statistics. In Table 2.3.5, the top selling Cryptopunks have an average price ranging from 39 to 62 ETH. Conversely, if we look at the least sold Cryptopunks in Table 2.3.6, the average price is around 5 ETH. Therefore, there may be a correlation between the number of sales of an NFT and its price.

Table 2.3.5: Descriptive statistics of the price of the 5th highest traded Cryptopunks in 2021

CryptoPunk ID	TX count	Mean	Std	Min	Max
5575	33.0	51.0412	34.3249	11.88	109.69
3856	32.0	56.2888	31.7866	15.000001	100.00
6913	32.0	45.9400	36.03409	14.5	120.00
7443	32.0	62.8019	41.9534	14.88	123.36
4130	31.0	39.3639	29.9469	9.549999	100.00

Table 2.3.6: Descriptive statistics of the price of the 5th lowest traded Cryptopunks in 2021

CryptoPunk ID	TX count	Price
1160	1.0	6.2700
3073	1.0	5.500
4998	1.0	4.6900
5909	1.0	4.6000
6129	1.0	5.1000

Looking at the top 5 selling crypto-currencies and the bottom 5 selling Cryptopunks in 2021, it may be interesting to know the attributes that make them up. The 5 best-selling Cryptopunks seem to have a majority of 2 Traits, while the least-selling ones seem to have a majority of 3 Traits.

Table 2.3.7: Analyse of the attributes for the 5 highest traded Cryptopunks in 2021

CryptoPunk ID	Attr 1	Attr 2	Attr 3	Attr 4	Attr 5
3856	Cap	Earring	nan	nan	nan
4130	Dark Hair	Purple Lipstick	nan	nan	nan
4601	Peak Spike	Classic Shades	Normal Beard	nan	nan
5575	Front Beard	Do-rag	nan	nan	nan
6085	Frumpy Hair	Mustache	nan	nan	nan

Table 2.3.8: Analyse of the attributes for the 5 lowest traded Cryptopunks in 2021

CryptoPunk ID	Attr 1	Attr 2	Attr 3	Attr 4	Attr 5
1160	Earring	Knitted Cap	Small Shades	nan	nan
3073	Bandana	Eye Mask	Big Beard	nan	nan
4998	Cigarette	Messy Hair	Frown	Chinstrap	nan
5909	Mohawk Thin	Regular Shades	nan	nan	nan
6129	Luxurious Beard	Do-rag	Regular Shades	nan	nan

2.3.3 Bored Ape Yacht Club analysis

The same analysis is made for the Bored Apes Yacht Club, visible on Figure 2.3.14. The trend in sales volumes follows that of the Cryptopunks with an increase around March and during the summer holidays. One can have an intuition that one market follows the other.

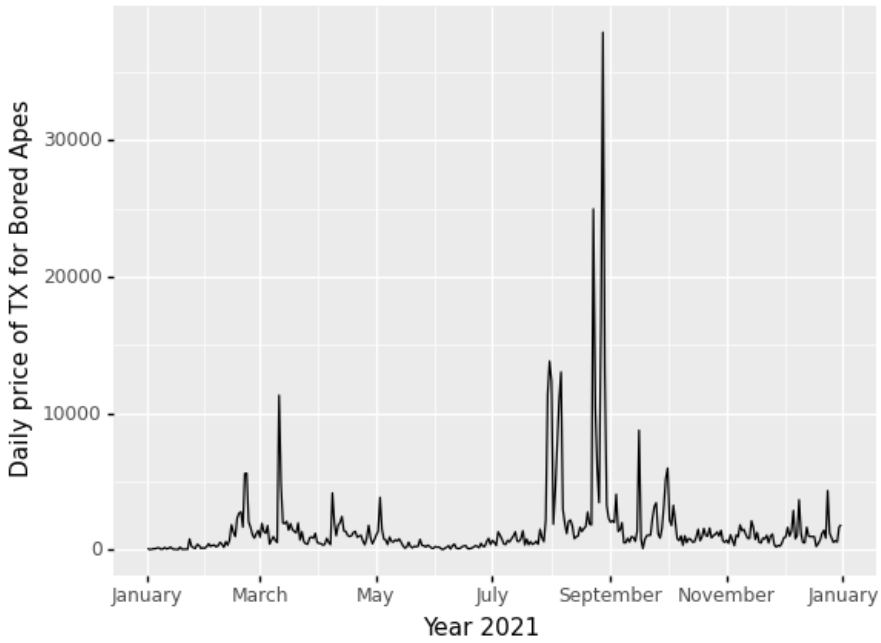

Figure 2.3.14: Daily price of TX for Bored Apes Yacht Club (2021)

Regarding the association of prices with the number of transactions in the tables below, unlike Cryptopunks, there is no big difference in the price of ETH price if BAYC was sold 10 times or 1 time, which implies that there may be no correlation between the price and the number of transactions of a certain NFT.

Table 2.3.9: Descriptive statistics of the price of the 5th highest traded BAYC in 2021

Bored Ape ID	TX count	Mean	Std	Min	Max
6788	12.0	27.458333333333332	18.890824532432447	4.75	55.25
3421	11.0	23.04181818181818	11.697064433593429	14.9	56.69
3914	11.0	37.35545454545454	28.16386811630688	5.49	100.0
4130	11.0	35.903636363636366	29.466362949209486	9.55	100.0
4601	11.0	37.703636363636356	30.167352310975943	14.95	99.0

Table 2.3.10: Descriptive statistics of the price of the 5th lowest traded BAYC in 2021

BAYC ID	TX count	Price
2149	1.0	25.97
2514	1.0	25.99
2665	1.0	30.0
3189	1.0	21.49
4456	1.0	75.0

2.3.4 Decentraland analysis

Decentraland is a virtual world of 300 x 300 plots for a total of 90,000 plots; this world is divided into districts and neighborhoods. It is difficult to imagine this world without a graphic representation. From the player's point of view, it is a 3D game in the third person as shown in figure . However, it would be more interesting to analyze the map of this world and see its characteristics.

Figure 2.3.15 shows the map of the Decentraland world. As said before, this is a a world composed of 90,000 plots. There are 9 main plazas called "Genesis Plaza" whose role is to welcome players when they enter the metaverse. When a player arrives in one of these plazas, a screen appears with the option to choose between 3 activities, playing classic games, participating in live events or accessing a list showing the location of other players. On a Genesis Plaza, there is also a main area with a bar where you can meet other players. However, it is also possible to buy plots in the Genesis. All the solid black lines represent roads that allow you to travel from district to district through the Decentraland. The blue plots are districts, of which there are 37. Districts act as independent communities led by a main leader and his coordination team. They have their own rules and their own universe. One district can be casino oriented as another one can be organized as a war game. Finally, the plots in green represent ones that are not governed by any community.

It may be interesting to track the price of the plot from its location on the Decentraland map, so that you can determine if there is a link between the price of the plot and being in a certain district, on a Genesis Plaza or near a road. Figure 2.3.16 and 2.3.17 give a better visualisation about this link.

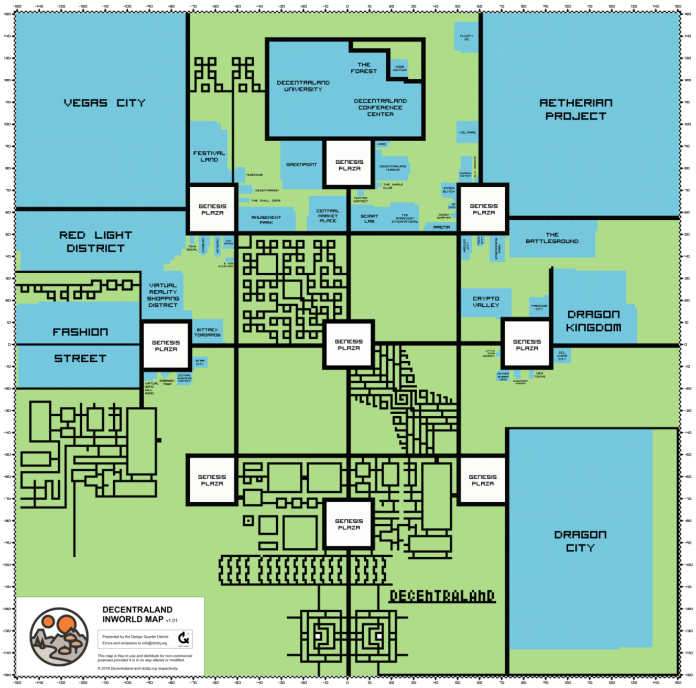

Figure 2.3.15: Decentraland map. Source.

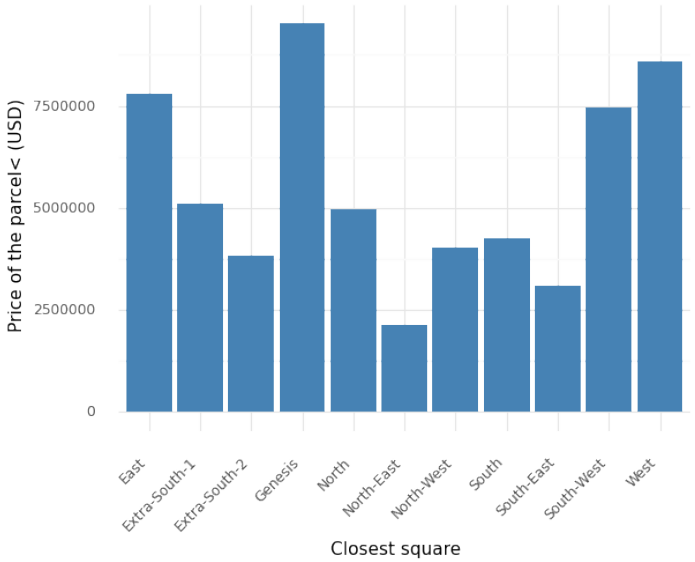

Figure 2.3.16: Histogram of the average price for different squares. It shows that some squares are more valuable than others, like Genesis, East, South-West, West square.

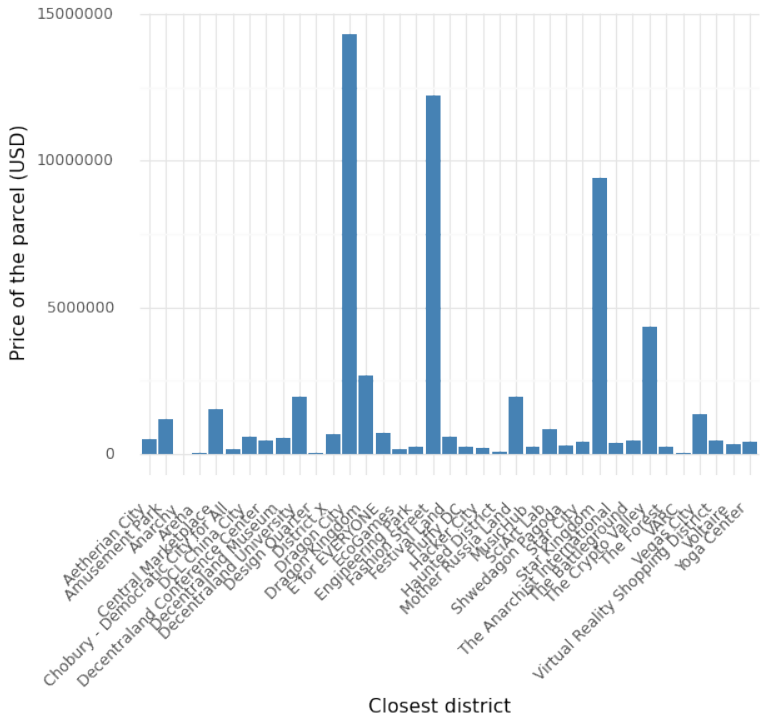

Figure 2.3.17: Histogram of the average price for different districts. It shows that some squares are more valuable than others, like Dragon city, Fashion Street or Star kingdom.

At last but not least, Figure 2.3.18 shows the daily price for all the transactions in the Decentraland world.

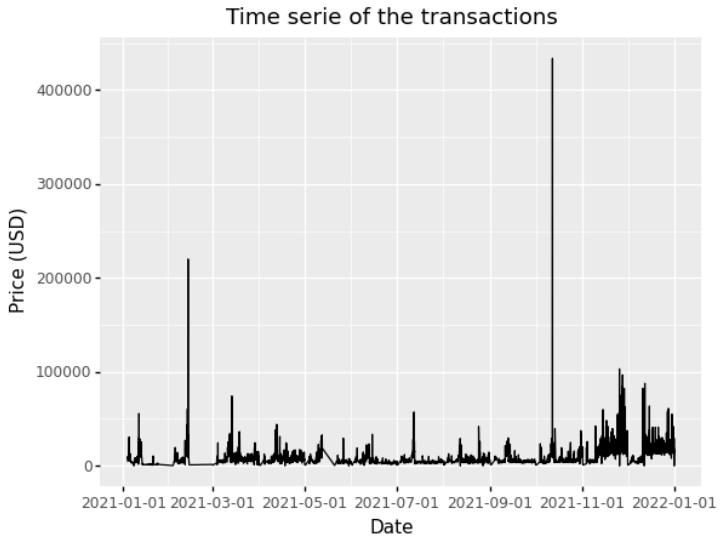

Figure 2.3.18: Daily price of the transaction in Decentraland for 2021

2.3.5 Twitter

Twitter is undeniably one of the largest social networks on Earth. It has an important impact on the world of NFTs as it is used as a medium to announce the launch of new collections and thus create a growing community. It also allows the management of a community gathered around the same interest for a certain collection. We can also notice another interesting aspect of Twitter: the biggest influencers with a large number of followers have a real impact outside the social network. For example, Elon Musk, one of the biggest scientists in the world today, who is followed by more than 77 million people, has an influence on the price of crypto-currencies just by publishing a tweet.

One is also able to generate a wordscloud in order to spot which words are the most often used in tweets. The bigger the word appears on the Figure 2.3.19, the most often it appears in tweets. The separate word clouds for positive and negative sentiments are to be found in the appendix.

Figure 2.3.19: Word Cloud of all tweets

.

Chapter 3

Predictive models conception

This section will be divided into several subsections to analyze the market in different ways and determine if any deductions and predictions can be made.

Here is a list presenting the threads that this section will follow:

1. Global predictive model creation of the price using a neural network (NN).
2. Analysis of the interest of Twitter sentiments on the price.
3. Predictive model creation for each collection using a neural network.
4. Global model creation taking into account the sentiments as well as the attributes of different collections.

3.1 Global predictive model creation of the price using a NN

Having a dataframe containing all the transactions for each collection with its price in US dollars, a first analysis would be to represent these transactions throughout the year to see the evolution of sales. Starting from the database containing all the transactions, each day D contains N transactions at different timestamps (Day, Month, Year, Minute, Second). To conduct the day by day analysis, we need to aggregate the data in order to group the transactions for day D only, without taking into account the hours and minutes. So we will aggregate the data by summing them by day.

$$Y_d = \sum_{i=1}^{n} X_{i,d} \tag{3.1.1}$$

where Y_d is the daily price, $X_{i,d}$ is the price of a transaction at day d.

To have a better visualization, we consider the logarithm of this sum. Figure 3.1.1 shows the result of a preprocessing of the data :

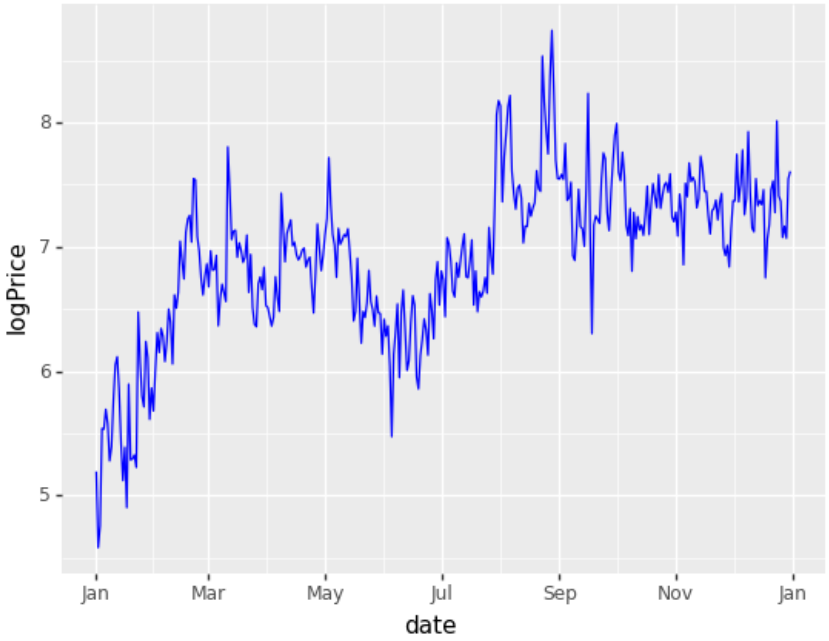

Figure 3.1.1: Time serie of the logPrice by day

The first part of this chapter consists in predicting this time series using a neural network. Neural networks are used in a variety of applications and their architectures can change greatly depending on the data being processed or the results expected. For the next subsections it is essential to present each specific architecture as it changes depending on the predictions we intend to make. The final dataset for this analysis is composed with two variables : Date and logPrice.

Unfortunately, neural network doesn't support input formats such as TimeStamp for the Date variable. It is essential to convert this date format into one that the model can understand. Since it is a time series that is analyzed here, one would like to know if there is a trend or a certain seasonality that can influence the model.

A priori, the model learns the trend and seasonality by itself, the deep learning models only take integers or floats as input. The question is: how to convert a date into a float ?

Dates can be converted into floats between -1 and 1. The neural network must understand that there is a cycle in the data. **Adams and Vamplew** [**1998**]

We want to preserve the cyclical nature of our inputs. One approach is to cut the datetime variable into three variables: year, month, day and then decompose each of those variables into two part. A sine and a cosine can be created as a facet for the month and for the week. For example, by creating the sine and cosine for the day, we want the model to

understand that Sunday is closer to Monday than Wednesday by taking the week. Following the same logic, we want the model to understand that December is closer to January than to April. The equation of the transformation are described in more details further in this thesis.

The date variable is present in preprocess as well as the logPrice variable. Indeed, if a neural network is fed by an input with too big values, the network will diverge and, as a consequence, learning will be impossible. To solve this problem, we have to rescale this variable so that it is between 0 and 1 which is acceptable by the network.

This thesis aims to address the problem through a Data Science approach. This approach allows to divide the data in 3 sets.

1. Train set: The model will be trained on this part of the data.
2. Validation set: A sample of data from the previous train set used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. This set is useful to trigger overfitting issue.
3. Test set: this set is used to evaluate the performance of the trained model on new data unseen before. It will be necessary to test the model on the data and compare the prediction with the label(s) already present.

In general, a first division in two of the dataset is done to have the training set and the test set for a proportion of 75% against 25% or 80% against 20%. Then a second division is made: the training set is split into a new training set and a validation set. The proportion of this split is the same as the previous one.

In the case of a time series, the data split cannot be random. The train set and test set data must follow each other in time. The available data cover one year from January 2021 to December 2021. Taking as objective to predict the month of December, the training set will be composed of all months except December. As a result, the test set will be composed of the month of December. This is a counter example to the split proportion given before. In this case, we will take a training set equal to 70% of the data, a validation set of 20% of the data and a test set of 10% of the data.

3.1.1 Architecture of the model

To predict time series data, the model will use a technique called sliding window. This window is composed of two parts: the inputs and the label(s). The idea is to use the previous inputs to predict the label(s). In the example below, each time t defines an hour. There are 24 times t in the inputs and 24 times t in the offset. This architecture allows to predict the 24 future hours based on the previous 24 hours.

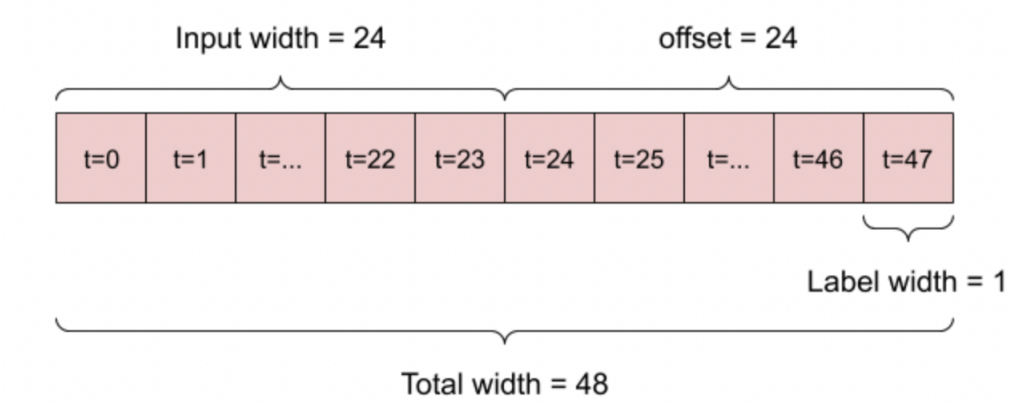

Figure 3.1.2: Sliding windows. Source.

The length of the offset can vary. The offset can be equal to 1 to predict the next hour, for example, based on the previous 24 hours.

The windows Generator tool in the tensorflow package allows to have a sliding window and to make the separation between input and offset very easily. As mentioned before, the number of timesteps to predict the future is a changeable variable. First of all, predictions of one timestep in the future will be undertaken, and then a comparison with predictions of 7 and 30 timesteps in time will be made. It is worth noting that a 7 time step represents a prediction for the whole week to come and the 30 time step makes a prediction for the month to come.

Now that the windows Generator is created, all that remains is to create a neural network to feed this network.

For time series, there are several neural network architectures that are feasible. We will create several of them in order to compare the metrics.

Four models have been selected in order to compare their predictive power.

- Baseline: This is the default model that gives the prediction of time t+1 in the future that is the same value as the actual one for the time t.

$$\hat{Y}_{t+1} = Y_t \tag{3.1.2}$$

- Linear: This is a neural network that only takes one layer with one neuron. This is equivalent to a linear model.

$$Y = aX_t + b \tag{3.1.3}$$

- Dense : This is a neural network that takes several hidden layers with, for each layer, an activation function " ReLU " and that finishes by an output layer with a neuron to predict a data in the future. This architecture is better detailed in section 4.2.5.2

$$Y = \phi(\sum_{i=0}^{n} w_i * x_i + b) \tag{3.1.4}$$

where: ϕ = "ReLU" activation function.
w = the weigths
x = the inputs
b = the bias

- LSTM: Otherwise known as "Long-Short term memory", it is a neural network that takes into account the price data of the past and assigns a weight to each of these prices according to its distance from the current data. A data at a time t depends on a data at a time t-1 and a data at a time t-2, ... ,data at a time t-10. However, the network puts a bigger weight on the time t-1 than on the time t-10. This type of neural network will be explained further in details.

After training these four models, the figure 3.1.3 shows the Mean Squared Error of the models, a first measurement made on the validation set during the training. After that, a second measurement is made on the test set to see if the model is fairly well trained. Both measurements are important to choose the best model. The Dense model is definitely the best in terms of training and testing because the loss on the validation and the test set are the lowest and the difference between the loss of the validation and test set in the Dense layer is not substantial.

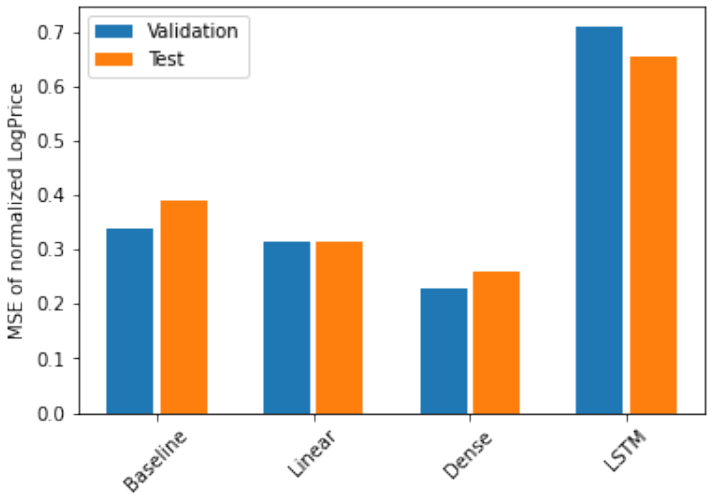

Figure 3.1.3: Performance of the model for diverse deep learning architectures

On the figure 3.1.4, the loss function for the train set and for the validation set are represented by the blue and orange lines. When learning a model, the goal is to minimize the loss of the train set. In the same way, the loss of the validation set will also decrease as a function of the training of the model. However, an overfitting problem can appear when the model learns too much on the training data, which results in the model not giving a good prediction on the test set. This problem is illustrated in the figure below. If the orange curve of the validation loss starts to increase after a decrease, this means that there is an overfitting problem. This can be seen in epoch number 45, the orange curve increases, which means that the model is no longer not learning correctly. Tensorflow allows you to restore the weights of a model before the overfitting so that the model is not affected.

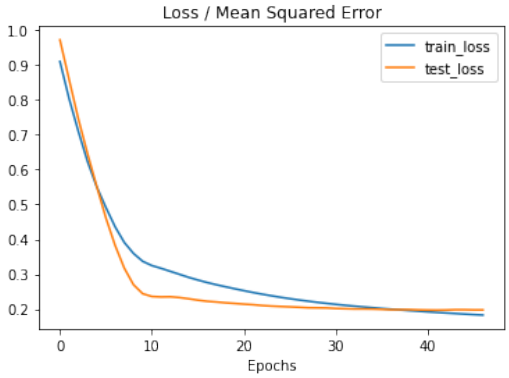

Figure 3.1.4: Loss and MSE of the model "Dense"

Now that the model is trained, it will be used to predict the month of December 2021 (more precisely from November 25 to December 31, 2021; that is to say 35 days). The figure 3.1.5 shows the orange line, representing the prediction for the last month of December overlaid on the real data. This prediction is obtained from the data of the previous months.

The figure 3.1.6 allows a better visualization by zooming in on the last month. We can notice that the model is visually quite good and fits rather well to the real data.

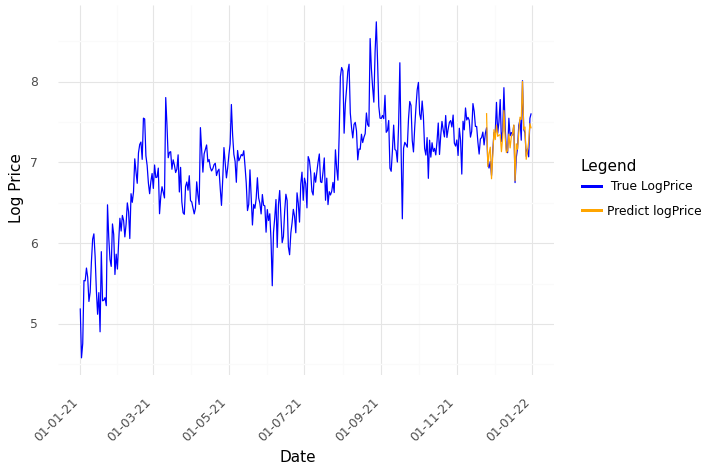

Figure 3.1.5: Prediction of the last 35 days of 2021

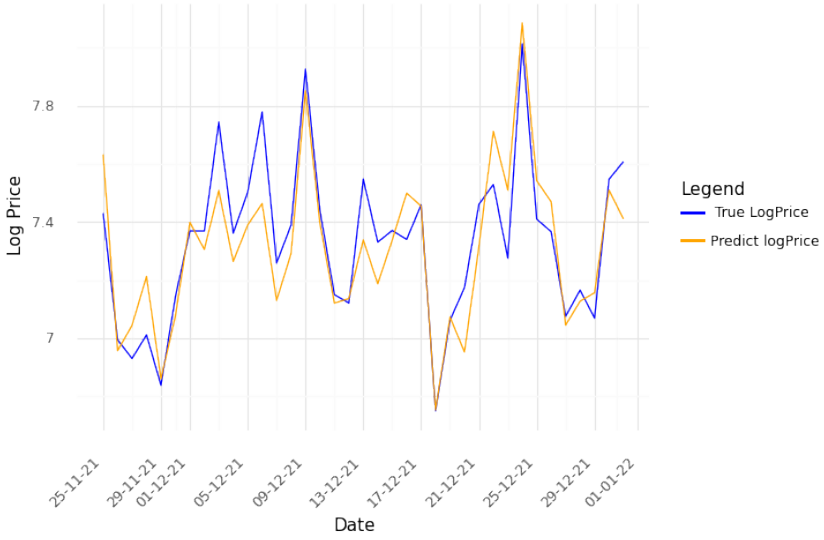

Figure 3.1.6: A deeper look at the month of December

Even if this graph is convincing, it is necessary to support the model with a metric that will quantify the model. This measure is defined by :

$$R^2 = 1 - \frac{Var(\hat{Y} - Y)}{Var(Y)} \tag{3.1.5}$$

where Y is the logPrice and $\hat{Y}$ is the prediction of the logPrice.

This measure is defined by "the goodness of fit" which is a value between 0 and 1.The closer the value is to 1, the better the model is and therefore the more reliable it is. The model give us a R^2 of **0.7765** which confirms the goodness of the model.

3.2 Analysis of the interest of Twitter sentiments on the price

Knowing that social networks can have a direct impact on the stock market and on cryptocurrencies, it would be interesting to do the same analysis with the NFT's market. This issue will be raised through a path composed of different steps.

1. Preprocessing of tweets.
2. Classification of each tweet in order to establish a sentiment for each of them. This part will focus on a new subject of data science that will be addressed: the "Natural Language Processing".
3. Transformations of the data in order to be able to compare the sentiments with the global prices of the NFTs.
4. Use of a statistical test to find out if one time series influences the other.

3.2.1 Preprocessing of tweets.

The composition of the dataset after the data collection via the Twitter API is in the following form:

date	followers_count	retweet_count	text
2021-01-01 11:53:19+00:00	2581	0	@HippoCartel did you send my hippo nft
2021-01-01 11:52:59+00:00	16444	1	RT @SuperRareBot: ð[U+009F][U+0092][U+008E] Arches ð[U+009F][U+0092][U+008E]\n\nâ[U+009C]¨ Artw...
2021-01-01 11:42:57+00:00	2361	213	RT @chainlinkangel: â[U+009C]¨ð[U+009F][U+008E]¨NFT GIVEAWAY!ð[U+009F][U+008E]¨â[U+009C]...
2021-01-01 11:40:05+00:00	469141	34	RT @MemeslotO: !Wish You A Happy New Year (202...
2021-01-01 11:35:29+00:00	27583	29	RT @Zaharia_af: ð[U+009F][U+0094]¥Have a look, this is your ...

Table 3.2.1: Elements retrieved from the Twitter API

From this database, the idea is to extract sentiments from the textual data of each tweet. Thus, the "text" variable is the one that captures the attention. It must be noted that the text is raw in the sense that it is not cleaned. The technique of natural language processing (NLP) will not work if it receives a text containing impurities as input. Here are the different steps followed in order to get a text ready to be used as input for our NLP model.

First of all, we have to remove the words that do not bring any value to the classification of words to extract a sentiment.

- Elimination of RT mentions: "RT @SuperRareBot:".
- Elimination of mentions of twitter users: "@: ".
- Elimination of numbers, emojis and figures.

In a second step, the tweets should be cleaned up by removing characters or words that are not essential or of a different alphabet than English, etc.

- Encode all the characters in ASCII.
- Use of a package to remove all words from a language different than English.

In a third step, it is necessary to convert all the characters to lower case because the NLP model for sentiment analysis is not case sensitive.

In a fourth step, we have to deal with what is called "Stopwords". These are words that are not useful for understanding a text such as "of, this, that, is, from, etc".

The last step is mandatory in order to be able to establish the algorithm of the sentiment analysis. It is necessary to tokenize the cleaned text beforehand in order to isolate each word.

After performing these steps, we get a cleaned text that we can find below.

date	followers_count	retweet_count	text
2021-01-01 01:12:01+00:00	17079	23462	[]
2021-01-01 03:46:21+00:00	8741	8	['show', 'best', 'cryptoa']
2021-01-01 06:24:39+00:00	11544	0	['yes', 'thank']
2021-01-01 06:25:29+00:00	18342	0	['haha', 'ainly', 'personality', 'thanks', 'hny']
2021-01-01 06:27:23+00:00	203206	8	['day', 'pranksyadvent', 'gave', 'joy', 'o', '...

Table 3.2.2: Twitter text cleaned.

As seen in the first line, sometimes there is no text left after cleaning because not all words make sentiment analysis useful.

3.2.2 Classification of tweets.

Now that the text is classified, we need to be able to predict a sentiment for it. There are already sentiment analysis packages that allow to produce a score for them. The Flair package will be used Akbik et al. [2019]; it allows to give a tag that corresponds to "Positive" or "Negative" according to the output of the NN used. After predicting for each sentence the sentiment, we obtain the following table and the following proportion:

Figure 3.2.1 gives a piechart representing the proportion of positive and negative tweets. The proportion of Positive/Negative tweets seems balanced.

date	followers_count	retweet_count	text	probability	sentiment
40327	2021-01-01 01:12:01+00:00	17079	23462	[]	POSITIVE
40358	2021-01-01 03:46:21+00:00	8741	8	['show', 'best', 'cryptoa']	POSITIVE
40357	2021-01-01 06:24:39+00:00	11544	0	['yes', 'thank']	POSITIVE
40356	2021-01-01 06:25:29+00:00	18342	0	['haha', 'ainly', 'personality', 'thanks', 'hny']	POSITIVE
13709	2021-01-01 06:27:23+00:00	203206	8	['day', 'pranksyadvent', 'gave', 'joy', 'o', '...]	POSITIVE

Table 3.2.3: Twitter text cleaned.

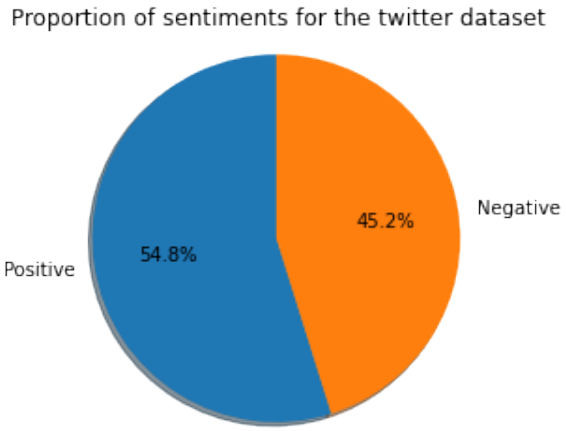

Figure 3.2.1: Proportion of sentiments

The score of each tweet relies on a pre-trained model called "en-sentiments", being one of the pretrained model contained in the flair package. The essential step to move from words to a score assigned to the corresponding sentence is to use words embedding.

Each word will be represented in a multidimensional space in the form of vectors. Let's say space is bidimensional, the good sentiment and bad sentiment are represented by vectors [1,0] and [0,1]. Taking the second line of table 4.2.3 as an example, the 3 words " show " , " best ", " cryptoa " can be represented in a 2D space; each word has a 2D representation in float. The result is calculated by taking the resultant of the three vectors and project this resultant on the Positive/Negative axis as represented on the figure 3.2.2 below.

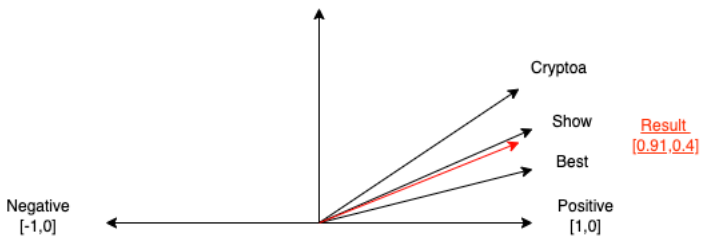

Figure 3.2.2: Illustration of Word Embedding

The explanation above is a basic concept of the word Embedding. However, in this paper, the suggested approach passes sentences as sequences of characters into a character-level language model to form word-level embeddings Akbik et al. [2018]. In order to understand this approach, it is necessary to introduce the concept of LSTM neural network in details.

LSTM Neural Network

When modeling textual information, a classical neural network becomes inefficient because it does not capture the spatial information that is essential for the modeling. This is where recurrent neural networks come into the picture. The difference with a classic neural network seen before is that the hidden layer constituted of one LSTM cell (also called LSTM neuron) will be looped on itself. As a result, each neuron has weights and receives the output of the previous neurons but also the value that came out of itself for the data t-1. The output of this same LSTM neuron will be used for the data at time t+1.

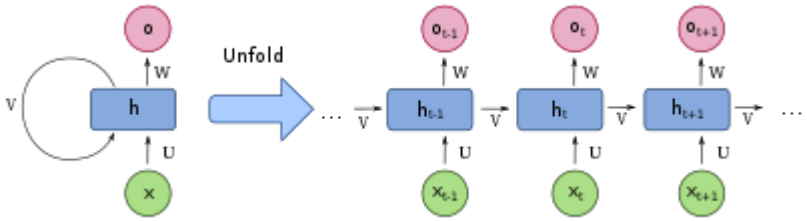

Figure 3.2.3: Representation of an LSTM neural network. X will represent the overall input, for instance, a sentence. Let's take the following sentence : "Today, it will rain a lot.". If the input x_t is represented by the word "rain", the words "will" and "a" represent the inputs x_{t-1} and x_{t+1}.

This architecture represents a short-term memory neural net, since every neuron retains information of the previous cell. However, the farthest a cell x_{t-i} is from the actual cell x_t, the less x_t holds information about x_{t-i}. Figure 6.0.4 in the Appendix represents this notion of retained information of the previous cells with time.

The gradient descent to adjust the weights is based on the following formula:

$$\omega_{new} = \omega_{old} - \alpha * F_{\omega} \tag{3.2.1}$$

where ω is the weight of a network
α is the learning speed of the network
F is the gradient of the network with respect to the weight ω.

The problem is that, by going to the left when updating the weights, the product that we subtract from the weight (alpha*F) becomes very small, which results in the quasi non-modification of the weights on the left of the architecture. Since every gradient depends

on the previous gradients, the gradient tends, in time, to a non-learning process or a loss of information, meaning the first layers don't really learn. This issue is known as the gradient dissipation problem or vanishing gradient. Hochreiter and Schmidhuber [1997], Sak et al. [2014a], Sak et al. [2014b]

And as a result, a RNN can forget old data when it learns. Its memory is a short-term memory. This is where the LSTM comes in, being an architecture capable of managing short-term and long-term memory. It will use a system of gates to regulate the information through the cells. The LSTM cell (Figure **??** is composed of three gates Smagulova and James [2019]. In order to have a short-long term approach, the set of operations present in a cell allows to keep or not to keep the information it has retained until now.

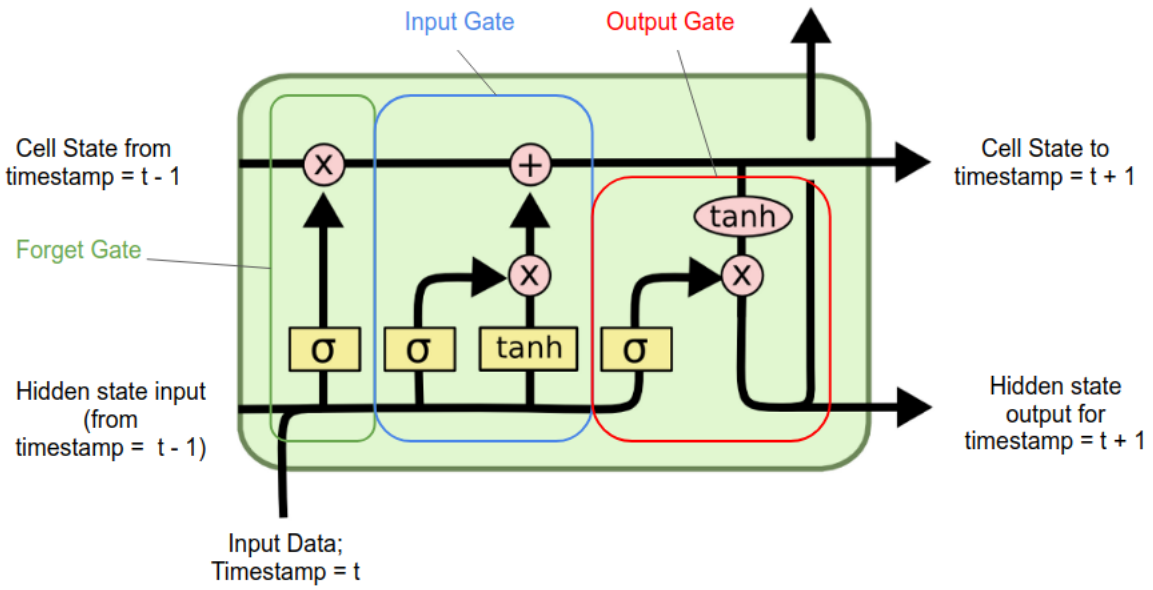

Figure 3.2.4: A LSTM cell. Source.

On Figure 3.2.4, inputs on the left part come from the previous LSTM cell. In the same way, the top-output is redirected to the next layer and the right-output reaches the next LSTM cell to give information. A LSTM cell is composed of five parts : three gates and two states Van Houdt et al. [2020]:

For every following formulas, the weight matrices (Wf, Wi, Wo, Wc, Uf, Ui, Uo, Uc) and biases (bf, bi, bo, bc) are independent of the time. Let's define each part :

Forget gate:

This gate can choose if it wants to keep the information or not. The information of the previous state is concatenated to the input data. A sigmoid is applied to the concatenation of both in order to normalize the data in a range [0;1]. The information is kept if the result is close to 1 and not kept if the result is close to 0.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{3.2.2}$$

where: σ_g = The sigmoid activation function
x_t = the inputs
h_{t-1} = The hidden state input

Input gate:

Its role is to extract information from the current data. The basic concatenated data are split and passed simultaneously to a sigmoid function (result between 0 and 1) and a tanh function (result between -1 and 1). The outputs are multiplied and will only keep the important information.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3.2.3}$$

where: σ_g = The sigmoid activation function
x_t = the inputs
h_{t-1} = The hidden state input

Output gate:

The output gate determines what output to generate from the current cell state. This operation is

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3.2.4}$$

where: σ_g = The sigmoid activation function
x_t = the inputs
h_{t-1} = The hidden state input

Cell state:

Long-term memory that stores and loads information from events that are not necessarily immediately preceding. The presence of the activation function **tanh** allows to prevent the vanishing gradient problem.

$$c'_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{3.2.5}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ c'_t \tag{3.2.6}$$

Hidden state:

Working memory ability, which carries information from the previous event and overwrites it uncontrollably at each step.

$$h_t = o_t \circ \sigma_c(c_t) \tag{3.2.7}$$

Tweets classifier

Now that Word Embedding and LSTM neural network are familiar, the classification is finalized with a Dense layer composed of a Softmax activation function to determine the polarity of the tweet. Figure 3.2.5 shows the model for the tweets classification.

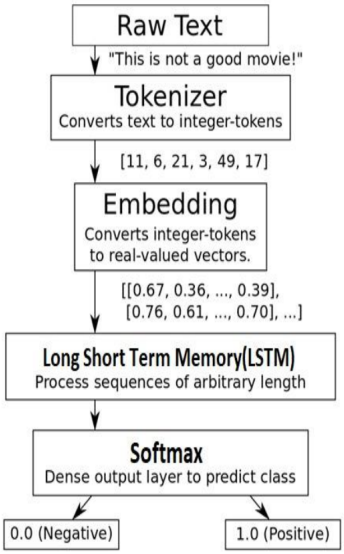

Figure 3.2.5: Representation of sentiment classifier Dr. G. S. N. Murthy [2020]

The Softmax activation function purpose is to normalize the outputs of the LSTM layer into a probability distribution, which sums up to 1.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3.2.8}$$

With this probability distribution, each sentence can be classified as "positive" if the probability is greater than 0.5 or "negative" if lower than 0.5.

3.2.3 Transformations of the data

Collecting only the sentiments doesn't do much to deduce the influence of them. They must be coupled with the number of followers of the sender of the tweet. Indeed, it is important to put a relationship between the sentiment of a tweet and the number of followers of the sender because this is what, if highly influencial, can perhaps change the state of the market of NFT's. Elon Musk's following tweet (Elon Musk has 77 million followers at the time of data collection, 102 million on August 3th 2022): "NFT's are the future, I'm going to invest 1 billion USD dollars in them", will have a much bigger impact than the same tweet posted by another NFT influencer with 150,000 followers. We will therefore create a sentiment score - SS - which will be in fact the sentiment score of a certain sentence multiplied by the number of followers of the sender of the tweet.

$$SS_i = F * S_i \tag{3.2.9}$$

où $S_i \in$ [-1,1] is the score for the tweet i, F is the number of followers of the writer of the tweet.

What is interesting is doing a daily analysis by comparing the daily price time series and the sentiment time series produced by the daily tweets. Of course, the two time series are on very different scales. Indeed, the sentiments are in the interval [-1; 1] while the daily prices are in the interval [3.807366e+04; 2.010891e+09]. We consider very different scales to see if the impact is really consequent. Another problem to take into account is that we have several sentiments on one day and only one price per day. In order to solve this problem, it will be necessary to carry out certain transformations. In this way, the sum of sentiment Index will be calculated for each day.

$$SS_d = \sum_i^n SS_i, d \tag{3.2.10}$$

3.2.4 Comparaison of the two time series.

Now, finally, the two time series will be compared day by day. We want to know if the feelings time series has an effect on the log price time series.

In order to compare the two time series, we want to introduce the notion of the Granger Causality test, a statistical hypothesis test for determining whether one time series is useful in forecasting another time series. Lima Cordeiro et al. [2020]

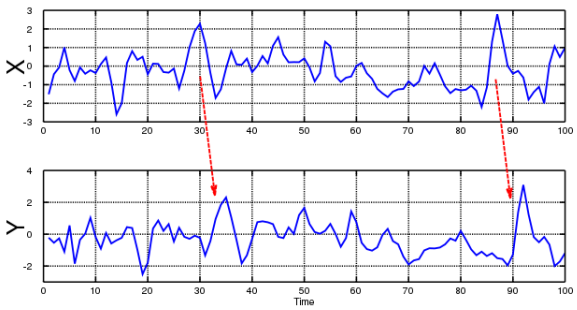

Figure 3.2.6: When time series Y (t) Granger-causes time series X(t), the patterns in Y (t) are approximately repeated in X(t) after some time lags (two examples are indicated with arrows). Thus, past values of X can be used for the prediction of future values of Y. Lima Cordeiro et al. [2020]

This test is essential to see the causality between a pair of time series and will therefore be used with both time series: log price and weighted sentiment. First, it is essential to visualize the data to see if there could be a potential impact. Second, it is important to consider that the impact will not be immediate. The influence may appear within a day, within 2-3 days or even within a week. This impact will be calculated. The figure 3.2.6 shows a pattern that allows to see this causality, the figure 3.2.7 below will do the same.

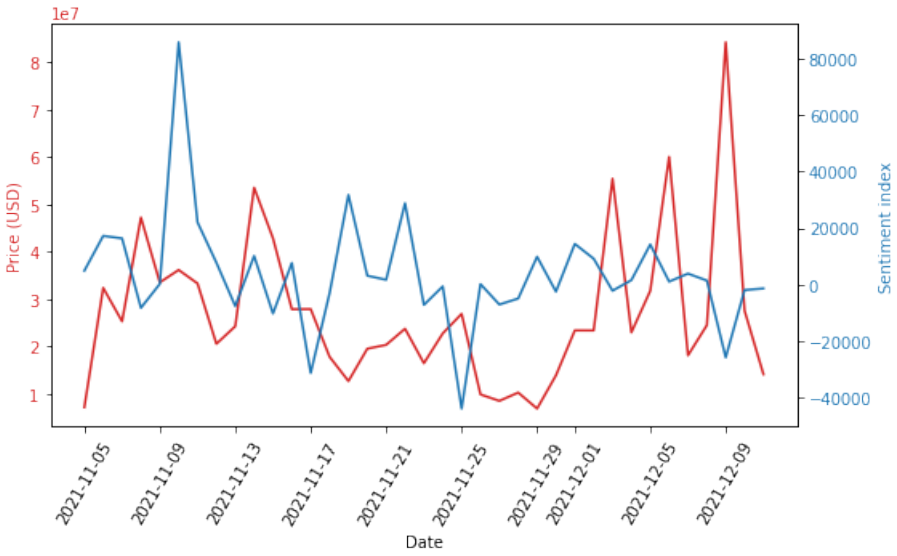

Figure 3.2.7: Comparaison of time series "log price" and "sentiment" for November 2021. You can find the all time serie visualization at Figure 6.0.5 in the Appendix

In figure 3.2.7, the red line represents the sentiment index constructed earlier and the blue line represents the sum of dollar sales of the overall market (i.e. including the collections described in the previous chapters). The plot focuses on the month of November 2021. The first sentiment peak, appearing on November 10, has an impact that is not immediate but visually reflected around November 14. The negative sentiment peak on November 25 generated a smaller sale the next day. All of these examples suggest that sentiments is influencing the market for NFT's.

Now that a visual example has been established, we need to support the facts with statistical methods to validate the previous intuition. For this purpose, the use of the Ganger Causality test is relevant.

In a more mathematical form, it is important to show the equations that derive from this test. This test assumes that the time series can be modeled from nth order autoregressive processes.

Here are 2 so-called AR time series.

$$X_t = \sum_{i=1}^{n} a_i X_{t-1} + \epsilon_t \tag{3.2.11}$$

$$Y_t = \sum_{i=1}^{n} c_i Y_{t-1} + \epsilon_t \tag{3.2.12}$$

A time series can depend on another by incorporating the other into its equation, which leads to the following equations:

$$X_t = \sum_{i=1}^{n} a_i X_{t-1} + \sum_{i=1}^{n} b_i Y_{t-1} + \epsilon_t \tag{3.2.13}$$

$$Y_t = \sum_{i=1}^{n} c_i Y_{t-1} + \sum_{i=1}^{n} d_i X_{t-1} + \epsilon_t \tag{3.2.14}$$

The test can be performed bilaterally. In this case, the test looks if X doesn't Granger-cause Y.

Granger causality test			
Model	**Regression**	**Hypothesis**	**Coefficients of X**
Partial	$Y_t = \sum_{i=1}^{n} c_i Y_{t-1} + \epsilon_t$	Null hypothesis	$\forall i, \beta_i = 0$
Full	$Y_t = \sum_{i=1}^{n} c_i Y_{t-1} + \sum_{i=1}^{n} d_i X_{t-1} + \epsilon_t$	Alternative hypothesis	At least one of $\beta_i \neq 0$

Let's define the hypothesis test, Y(t) and X(t) are two time series :

H0: X(t) does not Granger-cause Y(t)
HA : X(t) does Granger-cause Y(t)

This test must be carried out in a bilateral way where in the first case, Y(t) will be the time series of the daily sales prices and X(t) will be the time series of the feelings. Afterwards, the inverse operation will be made to check that the two time series do not influence each other. The objective is to have a one-sided effect, to see if the sentiments influence the market. After performing the tests, the table below shows the results.

Table 3.2.4: Bilateral Granger Causality test

(a) H0: sentiment does not Granger-cause price

Lags	F-test	P-value
1	3.7094	0.0549
2	5.3933	0.0049
3	5.0939	0.0018
4	5.4192	0.0003
5	4.3170	0.0008

(b) H0: price does not Granger-cause sentiment

Lags	F-test	P-value
1	0.2082	0.6484
2	0.7778	0.4602
3	0.1060	0.9565
4	0.1256	0.9732
5	0.4466	0.8157

In table (a), the p-value that it is lower than the 5% for lags > 1 which results in the rejection of the null hypothesis. It confirms that sentiments does Granger-cause price.

However, it is interesting to analyze the reverse situation and see if the price can Granger-cause the price. For that we can see in Table (b), that the p-value for all lags > 0.05, resulting in the non-reject of the null hypothesis saying that price does not Granger-cause sentiment.

3.2.5 Predictive model creation for each collection using a neural network

This sub-chapter is divided into 3 parts. The first one describes the different ways to process inputs depending on their type. The second part describes the architecture of the model (the layers and parameters used). The last part describes the results obtained for each collection.

Inputs

As a reminder, a neural network must take as input an array of numerical values. This input will pass through the layers of the neural network and will undergo a series of linear or non-linear transformations to finally emit an output which in our case must represent a numerical value estimating the log price of the NFT.

Questions emerge: for each collection, we will detail what the inputs are and how they have been transformed to feed the neural network.

For this description, a sample of the Cryptopunks collection will be taken as an example:

Depending of the types of your features, you need to apply some transformations to feed the neural net. In every category we have three types of variables : numerical, categorical and datetime variable.

Datetime variables

The date variable needs a special preprocessing to be a valid input for a neural network. The date has the following format "21-01-12", which is not possible to feed a NN. We must therefore take into account the temporal and cyclic aspects of the data.Adams and Vamplew [1998] We have to encode the data to make them cyclic and, afterwards, the neural network will learn by itself to detect a trend or a seasonality.

To handle this cycle, the data are transformed in this way:

Date
21-01-01
21-01-02
21-01-02

Date_year	Date_month	Date_day
21	01	01
21	01	02
21	01	03

month_sin	month_cos	week_cos	week_sin
0.2051	0.9787	0.7818	0.6235
0.2051	0.9787	0.97492	-0.2225
0.2051	0.9787	0.4339	-0.9009

Figure 3.2.8: Date transformation

The date is a timestamp containing the year, the month, the day. It can be interesting to capture information on the monthly and weekly cycles. Mathematically, we can transform the variables in this way:

$$\begin{aligned} month\_sin &= sin(m * (2\pi/month)) \\ month\_cos &= cos(m * (2\pi/month)) \\ week\_sin &= sin(d * (2\pi/(7 * day))) \\ week\_cos &= cos(d * (2\pi/(7 * day))) \end{aligned} \tag{3.2.15}$$

Numerical variables

Scaling is a good way to handle numerical variable having different scales. Indeed, if data are not scale, a divergence of the data can appear, that may be lead to a learning of the scale not being homogeneous. So convergence of the data is essential to obtain a viable solution. In order to increase the performance of your model, normalization is an almost mandatory step. Because updating the weights depends on the input value, gradient descent updates some weights much faster than others. This makes it more difficult to converge to the optimal value. Therefore, a Min-Max normalization is applied to each numerical data

to obtain values laying into the range [0,1]. The general formula for a min-max of [0, 1] is given by :

$$x_{normalized} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.2.16}$$

where: $x_{normalized}$ = the normalized value x.
x = the value x before the normalization

Even with variables which are scales, we can have problems of convergence, that is why it will be essential to use in its network a layer, called batch normalization, which will take care of rescaling the data in the middle of the learning process. This layer will be detailed in the next section.

Categorical variables

Textual variables cannot be fed to a neural network because it only accepts numerical data. We must therefore consider a transformation. The most basic transformation in machine learning is called the One Hot Encoder.

The principle of the OneHotEncoder is to associate an integer value to a category of a categorial variable. For example, for three categories, we would get 3 values 1, 2 and 3. This is an efficient method but in the case of a categorical variable with many categories (high cardinality), it can be a problem. The drawback of this method with high cardinality is that the resultant dataset is wide and sparse. In this case, it will need a huge amount of memory and increase the complexity of the model and the trend to overfit increase. The appearance of a concept called Embedding, solves this problem of high cardinality. As explained in the previous section, in an Embedding layer, the categories will become floats. These floats represent vectors in a certain space. For more theoretical details, see section 4.2.2.1.

Architecture

The architecture of a neural network is composed of different layers that are also called hidden layers. It is during these layers that transformations are applied to the data. These transformations are mainly basic mathematical operations.

- Input layer

  This layer is responsible for converting the inputs into a tensor. The tensor is the basic unit in a neural net and is a type of variable that is accepted by the neural net. In this way, the neural network is able to take as inputs, data of dimensions greater than 3.

- Dense layer

  Dense layer is a hidden layer composed of k neurons. Each neuron implements the operation:

$$output = \phi(\sum_{i=1}^{n} w_i x_i + b) \tag{3.2.17}$$

  where: ϕ = the activation function. This function can be linear, part-linear or non-linear.
  w_i = the weights (initially, there are created randomly).
  x_i = the inputs of the neural network.
  b = the bias. The value of the bias stay the same for each hidden layer.

  In a more visual way, Figure 3.2.9 shows the different parts.

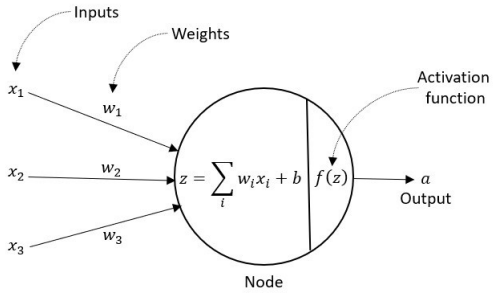

Figure 3.2.9: Dense layer

  So, the hyper-parameters for the model are the number of neurons in the Dense layer and the activation function to choose in each neuron.

- Embedding layer

  Explained in section 4.2.2.1.

- Concatenate Layer

  The concatenation layer takes care of concatenating the different layers. Let's imagine that we have two layers X and Y, the concatenation is defined by :

$$\underset{i\times k}{Z} = \underset{i\times j}{X} \times \oplus \underset{j\times k}{Y} \tag{3.2.18}$$

  where: Z = The concatenate tensor
  X = Tensor X from an embedding layer.
  Y = Tensor Y from an embedding layer.

- Flatten layer

  In order to feed the model, the result of the concatenation of the previous tensors must be reshaped. The Flatten layer allows to change the dimension of the tensor to have a tensor with only one column.

- Batch Normalization layer

  The Batch Normalization layer makes it possible to rescale the data taken as inputs of each layer in order to make the learning faster. Ioffe and Szegedy [2015].

  If the neural network uses a non-linear function as an activation function. Consider a layer with a sigmoid activation function z = g(Wu + b) where u is the layer input, the weight matrix W and bias vector b are the layer parameters to be learned, and $g(x) = \frac{1}{1+exp(-x)}$. As |x| increases, g′(x) tends to zero. This means that for all dimensions of x = Wu+b, except those with small absolute values, the gradient flowing down to u will vanish and the model will train slowly. However, since x is affected by W, b and the parameters of all the layers below, changes to those parameters during training will likely slow down the convergence. This effect is amplified as the network depth increases. This problem is known as internal covariate shift.

  This issue can be handled by using a part-linear activation function like ReLU or, if the keeping of a nonlinear function is essential, the Batch Normalization is the answer. This normalization is applied in three steps :

  1. Calculation of the batch mean and variance.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.2.19}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{3.2.20}$$

  2. Normalization of the layer inputs using the batch mean and variance.

$$norm = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.2.21}$$

where: ϵ = An arbitrary small constant for numerical stability.

  3. Scale and shift in order to obtain the output of the layer.

$$y_i = \gamma\, norm + \beta \tag{3.2.22}$$

where: γ, β = The parameters are learned during the training

One of the solution is the ReLU activation function. This function is defined by :

$$f(x) = \begin{cases} x & , x > 0 \\ 0 & , x <= 0 \end{cases} \tag{3.2.23}$$

The derivative of ReLU is 1 for values greater than zero. This basically solves the vanishing gradient problem since we still get 1 by multiplying multiple times. The negative component of the ReLU function is 0, so it is indistinguishable. Therefore, the derivative of negative values is simply set to 0.

- Dropout layer

  The dropout layer allows links to be dropped between neurons in two adjacent layers. This layer contains a parameter which is a real number between 0 and 1 and which represents the probability of dropping p neurons. The dropout allows the network to protect itself against overfitting. Multiple dropout techniques are out now, however, the focus is put on the standard dropout used in this paper. Mathematically, the probability of omission for each neuron follows a Bernoulli distribution of probability. It is followed by an element-wise multiplication between the vector of neuron (layer) with a mask in which each element is a random variable following the Bernoulli distribution. Represented mathematically, this concept is given by the formula :

$$y = \phi(W * X + b) \odot m, m_i \sim Bernoulli(p) \tag{3.2.24}$$

The output of the neural network is a float value that represents a prediction of the price of the NFT based on these features. Figure 3.2.10 shows the final model representation. The Concatenate layer takes ten Input layers, representing the categorical variables. The other Input Layer, next to the Flatten layer, holds a dimension (n,4) representing the Month_cos, Month_sin, Week_cos, Week_cos. The Batch Normalization layer is not applied in this model thanks to the use of the ReLU activation function.

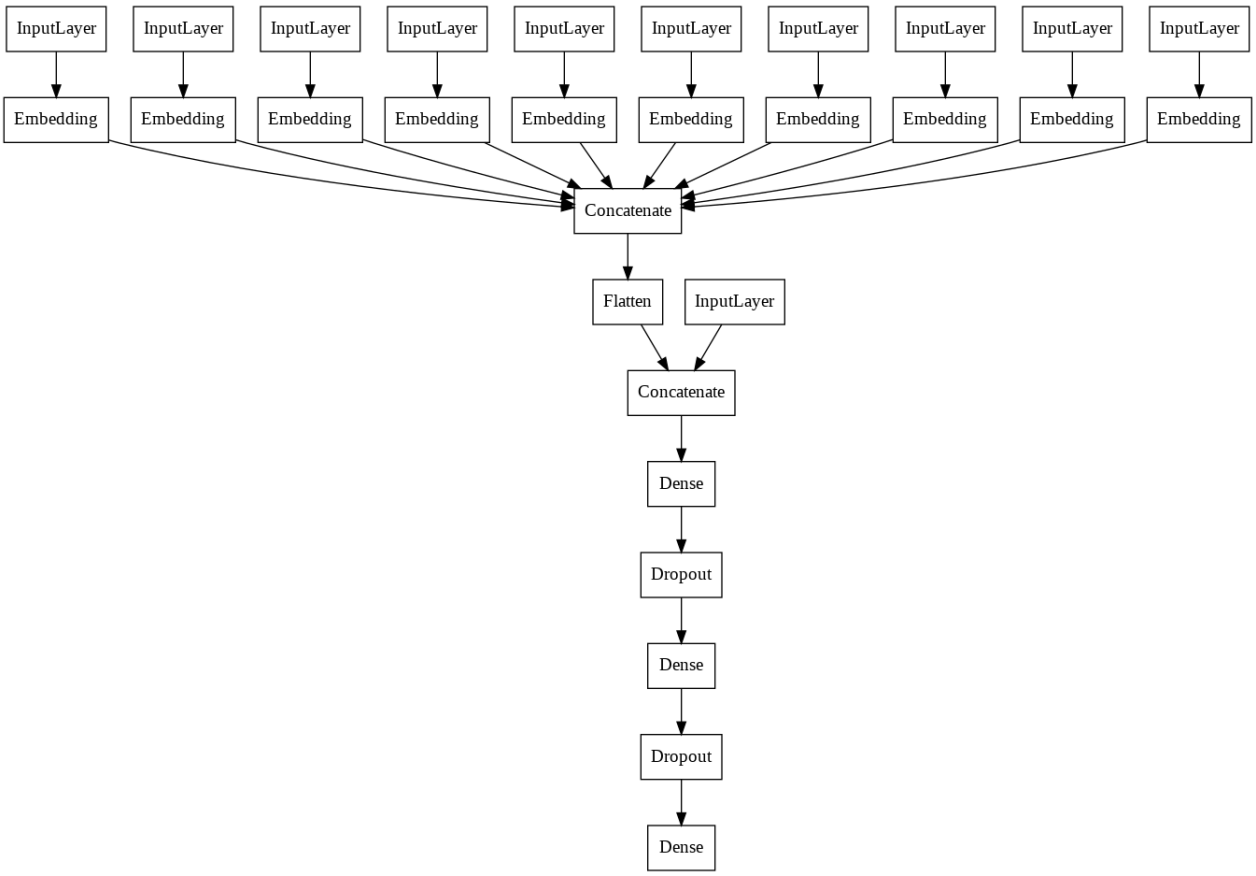

Figure 3.2.10: Architecture of the model for the Cryptopunk collection

A last step is to describe the hyperparameters of this model. Hyperparameters are internal to each layer and can influence the accuracy of the model. Differents hyperparameters are the number of cells in a layer, the activation function, the number of hidden layers, the probability for a dropout layer, etc. After testing a range of hyperparameters, here are the final results for the model to test with each collection. Here are the different range of values :

Table 3.2.5: Hyperparameters values tested for each hidden layer

	Hyperparameters values			
Neurons number	16	32	64	100
Activation function	Sigmoid	ReLu		
Dropout prob	0.2	0.3	-	
Batch size	8	16	64	

To be able to test theses hyperparameters, it is mandatory to hold a GPU. There are not many hyperparameters values in table 3.2.5. However, learning the best models when testing all theses values took about 5-6 hours with the help of a GPU. After testing these parameters, the table 3.2.6 show the optimal hyperparameters values for the model of figure 3.2.10.

Table 3.2.6: Optimal hyperparameters values for the model

	Cells number	**Activation function**	**Dropout prob**
Dense layer 1	100	ReLU	-
Dropout layer 1	-		0.3
Dense layer 2	16	ReLU	-
Dropout layer 2	-		0.3
Dense layer 3	1	Linear	-

Results

After describing the architecture for the Cryptopunks collection and testing the previous model on all collections, a metric can be deduced by Formula 3.1.5 which gives us an idea of the accuracy of the model. This metric will be computed for each collections.

Collection	Loss	MSE	R^2
CryptoPunks	1.3798e-04	1.8127e-04	0.7841
Bored Ape Yacht Club	0.0118	0.0286	0.7229
Decentraland	0.0157	0.0146	0.3073
Rumble Kong league	0.0331	0.0242	0.5529

3.3 Creation of a global model taking into account the sentiments as well as the attributes of different collections

This last part expresses the finality of this thesis by creating a model gathering all the collections and allowing to determine the price of an NFT from the information of the one sold at a day D and the sentiment generated by Twitter at the same day D. As expected, the model quickly becomes complex because it takes in input four collections that perform poorly. The R^2 is **0.12** which is not a surprising result due to the fact that the input matrix takes into account many variables.

Features importance

Understanding why a model makes a specific prediction can be as important as making a specific forecast. Deep learning is powerful, however, due to its complexity, the understanding of the model becomes more blurred and the influence of the features on this model gets lost in the network. Because a neural network is not easy to understand, a simpler explanation model must be introduced: an interpretable approximation of the original model. the notation of the original prediction model to be explained is f and g the notation for the explanation model. It can be said f can have an explanation model that is a linear function of binary variables Lundberg and Lee [2017], this is called **Additive feature attribution methods**:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{3.3.1}$$

where: $z' = z' \subseteq \{0,1\}^M$.
M = The number of simplified input features.
ϕ_i = Shapley values for feature i.
ϕ_0 = The output of the null model meaning the mean output of the model.

Shapley values assign to each feature an importance value for a particular prediction. It is based on the concepts of game theory and can be used to explain the predictions of machine learning or deep learning model by calculating the contribution of each feature to the prediction. SHAP is an individualized model-agnostic explainer, meaning any model can be explained without knowing how that model internally works. In order to quantify the implication of each variable, Shap values needs to be explained. This concept comes directly from the game theory world.

Shapley value estimation method

Imagine a game where the players have to collaborate together (called a C coalition) to obtain a certain value V. It would be interesting to know what each player's contribution is to obtain the final value. A parallel can be found by taking the variables of a model as players and the output of the same model as V-value. The principle is to calculate the output of the model by subtracting a feature from the inputs Lundberg and Lee [2017] . Let's define F, the set of all the features (inputs) of the model. Shapley needs to retrain the model on all features subsets S of F where $S \subseteq F$. Two operations are needed to compare them. Train a model with a feature i (full model) and train the same model without the feature i (reduced model). The full model and reduct model prediction notation are given by $f_{S\cup\{i\}}$ and f_S. From these two predictions, we compute the difference for the current input $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$, where x_S represents the values of the input features in the

set S. This difference is computed for all possible subsets $S \subseteq F \backslash \{i\}$ since the deduct of a feature depends on other features of the model. Shapley values are weighted average of all possible differences given by the following formula:

$$\phi_i = \sum_{S \subseteq F \backslash \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \tag{3.3.2}$$

Additive Feature Attribution Methods ensure the existence of a unique solution due to three properties :

- **Local accuracy** : if the input and the simplified inputs are roughly the same, then the actual model and the explanatory model are roughly the same. Mathematically : if $x' \approx x$, $f(x) \approx g(x')$
- **Missingness** : if a feature is excluded from the model, its attribution must be zero. Mathematically : if $x'_i = 0, \phi_i = 0$
- **Consistency** : if a feature contribution changes, the feature effect cannot change in the opposite direction.

After calculating the shap values, Figure 3.3.1 for the final models the importance of the following features. As assumed, collection Cryptopunks and BAYC have a bigger representation than Decentraland and RKL with the features "Gender", "Attribute1", "Attributes number". Sentiments have a feature contribution of nearly 8%, meaning it is a consequent feature. Time is also a consequent feature due to the representation of the variables "Month_sin", "Month_cos", "Week_sin", "Week_cos".

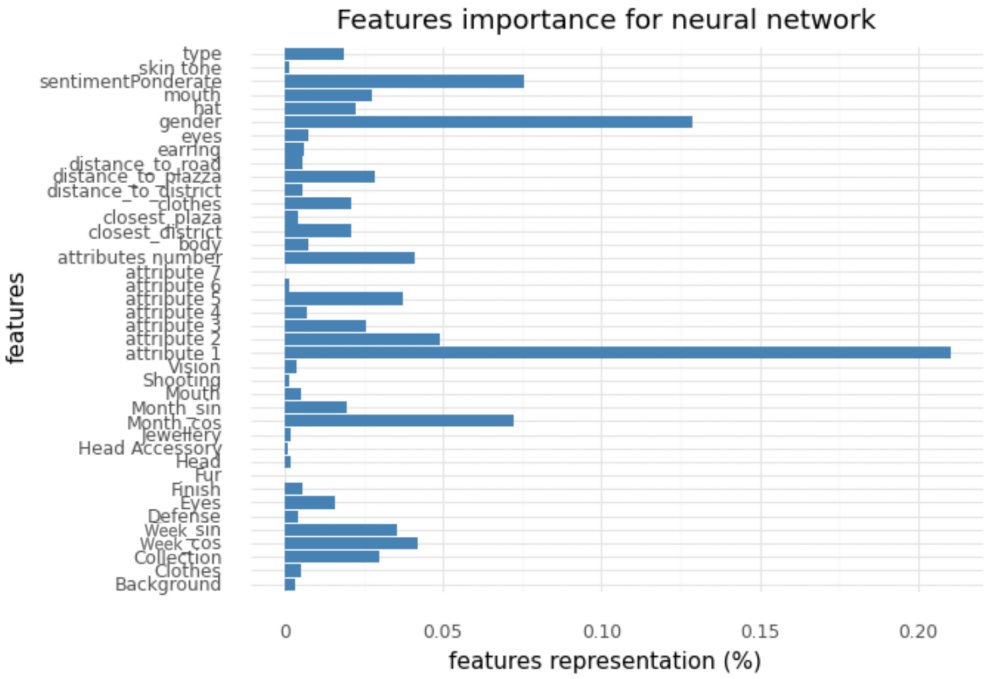

Figure 3.3.1: Features importance for neural network

Chapter 4

Conclusion

This thesis is filled with in-depth concepts that are very different from each other. The knowledge integrated during this thesis is monumental. Not knowing the world of crypto-currencies and NFTs before writing this thesis, I did not imagine all the knowledge to be gathered in order to understand this subject in great detail. The world of blockchain, crypto-currencies, and NFTs is vast, several months will have been necessary for me to learn how this new technology has reached the world of individuals, and generated a certain attraction towards investors in 2021. This dissertation gathers a lot of different parts of machine learning and deep learning, from sentiment analysis to neural networks, time series and natural language processing. It is a very broad topic where the goal of producing predictive models for different aspects of the market has been achieved. Some models, such as the global predictive model of the market of the main collections, or the models of the Cryptopunks or BAYC collections, have a promising future due to their accuracy which could hold in time. Other models, such as the Decentraland or RKL collections, perform in a less successful way. And finally, as expected, the global model, mixing the different collections with the associated feelings per day, is of mediocre quality because of its complexity.

It is always possible to improve a model by testing different values for the hyper-parameters as tested in this thesis. However the tests were brief, with a maximum number of 4 hyperparameters, because being limited by the computing power of the Google Colab GPU. An improvement track could mean the improvement of more hyperparameters with a larger range of values for each of them.

Another track would be the enhancement of NLP techniques in order to be able to define even better feelings. The preprocessing detailed in this paper is a basic preprocessing, even if more techniques of preprocessing and analyzing feelings exist. Every month, new academic papers are published, giving way to new possibilities for creating deep learning models that can lead to better performance. Data science is a constantly evolving science that is moving at a rapid pace. This same data science allowed the writing of this thesis.

Chapter 5

Acknowledgements

First at all, I would like to thank Mr. Christian Hafner, my supervisor, who supported me throughout the writing of this thesis by sharing his knowledge and his advices. These precious advices allowed me to question myself on some concepts of deep learning and to deepen my knowledge. This thesis has evolved a lot throughout starting from a slightly different idea of the final rendering through questioning the purpose of this thesis. I would also like to thank my parents who supported me in every way they could during the writing of this thesis and Roxane Deveen, who helped me proofreading. And finally, I would like to thank all the people who supported me in any way, without forgetting the platforms that allowed me to obtain the different data such as the different APIs of Twitter and Opensea.

Chapter 6

Appendix

Figure 6.0.1: Descriptive statistics about the skin tone and the number of attributes

Figure 6.0.2: Negative word cloud

.

Figure 6.0.3: Positive word cloud

.

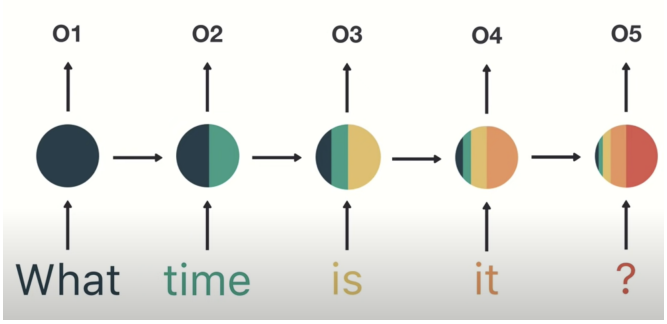

Figure 6.0.4: RNN architecture with a short-term holding of the information.

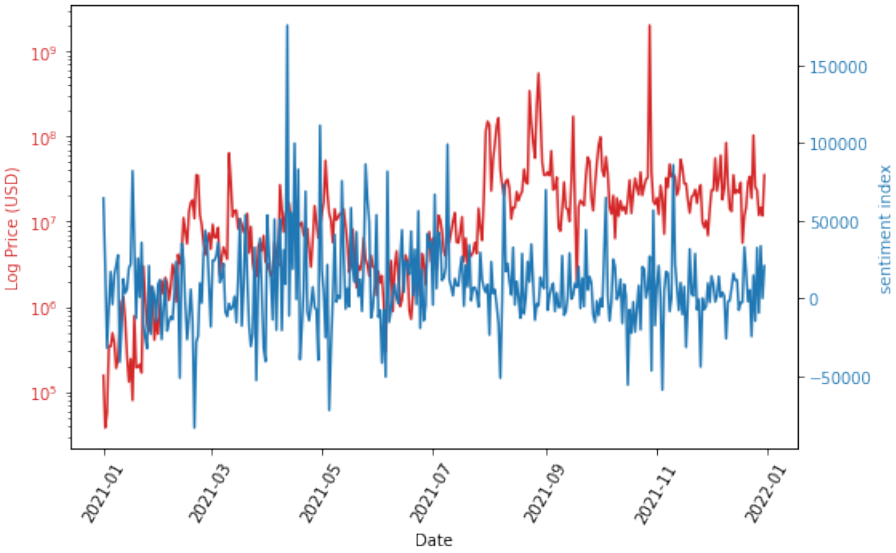

Figure 6.0.5: The logPrice and sentiments time series

Bibliography

A. Adams and P. Vamplew. Encoding and decoding cyclic data. *The South Pacific Journal of Natural Science*, 16, 01 1998.

A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

L. Ante. The non-fungible token (nft) market and its relationship with bitcoin and ethereum. 2021.

M. Belotti, N. Bozic, G. Pujolle, and S. Secci. A vademecum on blockchain technologies: When, which and how. *IEEE Communications Surveys Tutorials*, PP:1–1, 07 2019. doi: 10.1109/COMST.2019.2928178.

U. W. Chohan. Non-fungible tokens: Blockchains, scarcity, and value. 2021.

P. Cuffe. The role of the erc-20 token standard in a financial revolution: the case of initial coin offerings. 2018.

B. A. M. B. M. B. Dr. G. S. N. Murthy, Shanmukha Rao Allu. Text based sentiment analysis using lstm. 05 2020.

W. Entriken, D. Shirley, J. Evans, and N. Sachs. Eip-721: Erc-721 non-fungible token standard. *Ethereum Improvement Proposals*, (721), 2018.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9: 1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

U. K. Khalid Husain Ansari. Implementation of ethereum request for comment (erc20) token. 2020.

V. Lima Cordeiro, F. Dellajustina, R. Shimoura, M. Girardi-Schappo, N. Kamiji, R. Pena, and A. C. Roque. Granger causality in the frequency domain: derivation and applications. *Revista Brasileira de Ensino de Física*, 42:e20200007, 09 2020. doi: 10.1590/1806-9126-RBEF-2020-0007.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

F. D. G. M. M. L. M. A. Matthieu Nadini, Laura Alessandretti and A. Baronchelli. Mapping the nft revolution: market trends, trade networks, and visual features. 2021. doi: 10.1111/j.1467-9892.2007.00537.x.

M. Mazur. Non-fungible tokens (nft). the analysis of risk and return. 10 2021.

Y. L. Nicola Borri and A. Tsyvinski. The economics of non-fungible tokens. 2022.

A. Porat, A. Pratap, P. Shah, and V. Adkar. Blockchain consensus : An analysis of proof-of-work and its applications. 2017.

H. Sak, A. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 338–342, 01 2014a.

H. Sak, A. Senior, and F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014b. URL https://arxiv.org/abs/1402.1128.

G. G. Shahar Somin and Y. Altshuler. Erc20 transactions over ethereum blockchain: a network perspective. 2020.

K. Smagulova and A. James. A survey on lstm memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228, 05 2019. doi: 10.1140/epjst/e2019-900046-x.

A. Toygar, C. E. T. Rohm, and J. Zhu. A new asset type: Digital assets. *Journal of International Technology and Information Management*, 22:7, 2013.

G. Van Houdt, C. Mosquera, and G. Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020. doi: 10.1007/s10462-020-09838-1.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des Sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/sc