Analyzing and Modeling NFT Price Behavior

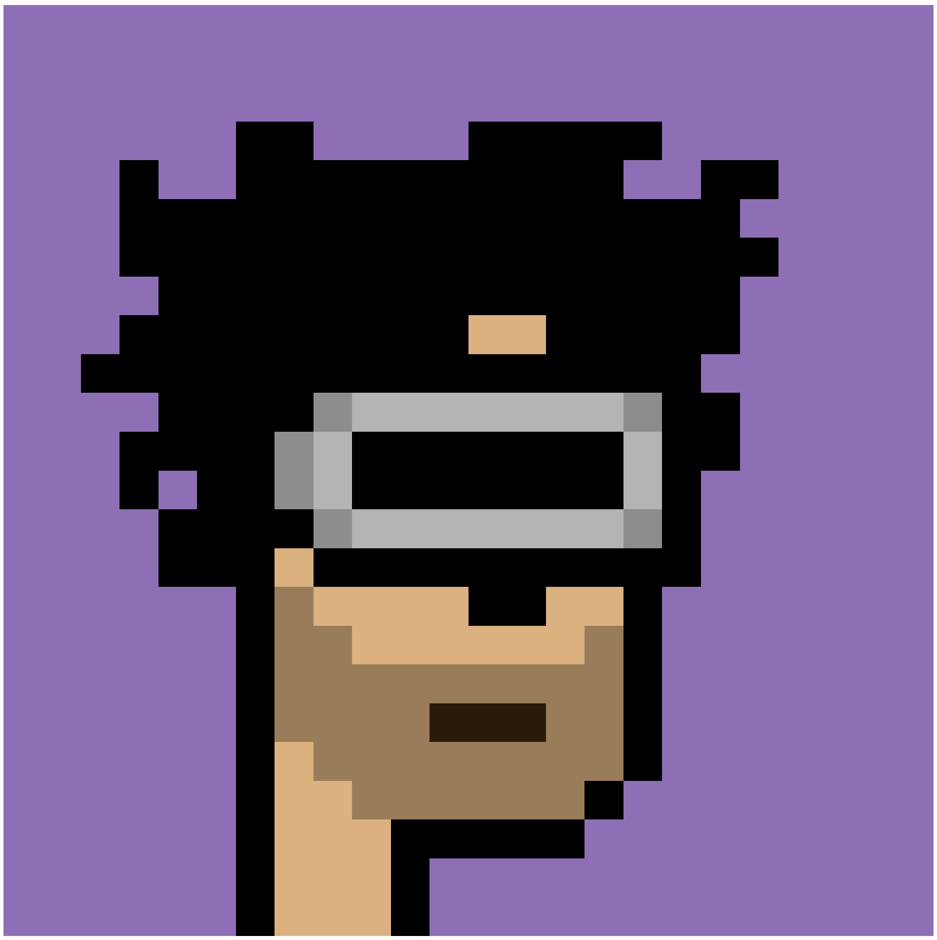


Figure 1: CryptoPunk #8102

Source: [1]

Team Stonks

12.12.2021

IEORE4523 Group Project | Fall 2021

GROUP MEMBERS

- Monica Guntur (mg4363 | Section 3)
- Clemente Domeyko (cd3255 | Section 3)
- Hsu-Sheng Ko (hk3176 | Section 3)
- Kazuhiro Matsumoto (km3711 | Section 1)
- Tarusha Silva (trs2158 | Section 1)

PROJECT DESCRIPTION

Project Goal

To build a model that uses attributes of non-fungible tokens (NFTs) and their market history to predict future market prices in order to develop an investment strategy.

Project Scope

Analyzing and developing a model for an NFT collection called “CryptoPunks” that spans 10,000 tokens developed by Larva Labs [2]. They represent ~40% of the market capitalization of NFTs and rank amongst the top sales every month [Figure 1 and Figure 2 in Appendix A], indicating relatively more frequent and recorded transaction data than other NFTs.

INTRODUCTION

Description of NFTs

NFTs represent a unique digital item that is stored on a blockchain [3].They can represent anything digital, including art, music, in-game purchases, and tickets [4]. Non-fungibility separates it from a cryptocurrency, where trading one NFT for another will always result in each party having assets that are very different to each other [4]. Most NFTs are stored on the Ethereum (ETH) blockchain enabled by the ERC-721 standard that enables various functionalities like token transfer protocols and ownership tracking [5]. Smart contracts developed by NFT creators and cryptocurrency wallets held by investors facilitate market transactions [5].

Overview of NFT Markets

2021 is seeing a surge in NFT trading activity, especially in the periods after the first half of 2021 [6]. In Q3 2021, overall market transactions grew by 382× y/y [Figures 3 and 4 in Appendix A]. Primary sales, secondary sales, and USD amount spent on NFTs (aggregated weekly) have also risen significantly in 2021 from lackluster levels in 2020 [Figures 5, 6 and 7 in Appendix A]. ~40% of NFTs that are sold can be categorized as collections, to which CryptoPunks could be categorized into and most NFTs trade in the USD 101-USD 1,000 range [Figures 8 and 9 in Appendix A].

PROJECT DATA

Data Sources

We gathered data on the CryptoPunks NFT collection and its historical market transactions using the API of an NFT marketplace called OpenSea [7]. OpenSea dominates NFT transactions that take place on the ETH blockchain [8]. To convert the ETH price of NFTs from OpenSea, we used the ETH/USD exchange rate data from Yahoo Finance [9]. Moreover, historical market transactions data were supplemented with additional data on the market conditions prevailing at the time of the transactions from NonFungible.com [10] in order to improve our model along with Twitter data to gauge market sentiment on NFTs at the time of the transactions.

Data Interfaces

The following interfaces were used to gather the data:

- OpenSea API (only way to get the data as web scraping is disabled by the company)
- pandas_datareader
- snscrape (could not use tweepy as Twitter API limits data to the most recent 7 days)

Overview of Data

Types of Data Gathered:

- Asset Data: *Data on the attributes of each token (e.g. type, accessories, etc.)*
- Event Data: *Historical data (from 2017 to Nov 2021) for each token (e.g., sale price, bid, etc.)*
- FX Data: *Historical ETH and USD exchange rates*
- Market Condition Data: *Collection of non-price related market data (e.g. active market wallets, number of primary sales, etc.)*
- Text Data: *Collection of tweets on CryptoPunk*

Describing CryptoPunks

CrypoPunks can be categorized into 5 main types and there are 87 attributes (other than type) that each token can take, with each CryptoPunk having either zero or a maximum of 7 seven attributes (other than type) in total [Figures 1,2 and 3 in Appendix B]. Each token is randomly generated and inherits these unique properties with relative rarity. Rarity is not available as a data point from OpenSea API, but it can be calculated using a variable we get as 'trait_count', which indicates how many other tokens have the same attribute as a particular NFT [Figure 4 in Appendix B]. We assigned a score to each NFT based on its attributes and relative rarity [Figure 6 in Appendix B] using the formula below:

NFT Score = SUM(1 / [Number of tokens with trait / Total number of trait type in collection]) [12]

A higher score indicates more rarity, and we expect more rare tokens with higher scores to fetch higher prices. A mean rarity score of 250 was observed for our data set [Figure 8 in Appendix B]. However, a plot of USD against token scores did not show any strong relationship [Figures 9 and 10 in Appendix B]. A plot of Twitter sentiment against ETH prices also did not provide any indication of a strong relationship [Figure 11 in Appendix B] to support including it as a factor in our model to determine token prices.

Data Cleansing

There was a lot of missing data in the Event Data set. To overcome this, we focused our analysis on transactions categorized as 'successful' by OpenSea (these represent sales transactions). For those observations without successful transactions, we used max_bid and/or min_ask as a proxy for the price. The benefit of this approach is that we significantly increase the number of observations, but at the cost of the assumed price being less informative than the price of an actual successful transaction. Moreover, there were several outliers in the Event Data that had to be manually rectified [Figures 12 and 13 in Appendix B].

MODELS AND RESULTS

K-Means

We utilized the K-Means algorithm in an attempt to group tokens by their attributes and determine their inherent value without historical price as a feature. Our hypothesis was that since some attributes are rarer than others, tokens with the same rare attributes will be grouped in the same cluster and that cluster will have a higher average price than others. The data was transformed using one-hot encoding to separate all attributes into their own individual columns with “1” in cells to indicate whether a token has that attribute. We utilized the elbow method to determine the optimal number of clusters. After running PCA to visualize the clustering result, we saw that even though there seemed to be several distinct clusters, there were still a large number of misclassified tokens. The average price of each cluster seemed to be different [Figure 1 in Appendix C] but the average score of each cluster [Figure 2 in Appendix C] was all similar.

K-Means with NFT Score & PCA

Instead of using one-hot encoding, we tried to utilize NFT scores for K-Means. Also, because there are many features, we applied PCA to reduce the dimensionality.
To do so, first, we transformed the raw data [Figure 6 in Appendix B] into a pivot table [Figure 1 in Appendix D] (an index represents a token_id, a column represents an attribute and a cell represents a NFT score). Then, we standardized the values (subtracting the mean from the scores, then dividing it by the standard deviation), and performed PCA to calculate Principal Component Scores for each token [Figure 2 in Appendix D]. Finally, we picked up the first 58 Principal Components out of 92 based on the cumulative contribution (which is greater than 70%) [Figure 3 in Appendix D].
Based on the graph [Figure 4 in Appendix D], we decided to apply K-Means with K = 2 to Principal Component Scores. Then, we grouped tokens by attributes for each cluster, and counted the number for each attribute. Clearly, it’s divided by gender [Figure 6 in Appendix D]. However, as we increased the number of clusters, the leading attributes for each centroid became less distinct and did not show the results we were expecting. With k=11, we saw two distinct clusters (6 and 8) with higher average prices, but their average scores were on par with the rest of the clusters [Figure 5 in Appendix D].

Predicting Token Price Movement using Machine Learning Classifiers

Feature Selection

For the classification task, we chose a data format (see Figure 5 in Appendix F). By ordering the attributes from attribute0-attribute6, we ensure that tokens which have similar attributes in similar positions are grouped together. The other features are the number of attributes and last year’s price movement which is a numerical and categorical variable respectively. With the input data, we predict the label for each row as the price movement for the current year. This is a multinomial classifier, with the labels denoting if the price moved up (**profit**), moved down (**loss**) or didn’t move at all (**nil**). After the feature extraction, we process the data for the machine learning pipeline. Categorical variables are one-hot encoded and the numerical variables are scaled.

We chose 2 basic classifiers to understand if our approach could result in meaningful results. Thus we started with a basic classifier - Logistic Regression and moved to one of moderate complexity - Gradient Boosted Classifier. Our methodology for the experiments were as follows -
1. Fix a particular year as the prediction set (holdout data), in this case - 2020
2. Use the previous year’s data to train and test the classifier, in this case - 2019. We perform cross-validation using accuracy as the scoring metric.
3. Use the trained classifier to make predictions and check metrics i.e. use the classifier fitted on 2019 data to make predictions for 2020 and test against actual data

Logistic Regression

Logistic regression[15] was the first classifier we tested. Since the data is unbalanced, we also tested the classifier by adjusting the class weights as the inverse frequency of the class (in-built feature of scikit-learn). With this, we reported an accuracy of 82% and 62% for the weight unbalanced and balanced cases. Even though the accuracy is better in the first case, we noted that due to most inputs having no sales, it predicts every token to not have any price movement, thus resulting in high accuracy, but large number of misclassifications of profits and losses (see Figure 1 and Figure 2 in Appendix F). This is reflected in the Receiver Operating Characteristic Area Under Curve (ROC-AUC)[16] score which is 0.71 and 0.81 respectively. Thus as a first study, we note that attributes and previous price movement are useful for predicting current year price movement. We also modified the threshold parameter to introduce higher bias in the classifier. This reduces the mis-predictions of losses as profit but biases the predictor to predict loss for every input (see Figure 3 in Appendix F)

Gradient Boosted Classifier

Given that the data is imbalanced and mostly categorical features are used, we decided to use an ensemble method - Gradient Boosted Classifier[17]. Since ensemble methods train multiple decision trees on the data and use the weighted outcome of each method for the final prediction, class imbalance is handled by the ensemble. Following the same methodology as above, we trained this classifier and made the predictions. We reported an accuracy of 81% and the confusion matrix (see Figure 4 in Appendix F)and the ROC-AUC metric is 0.41. Thus, even though the accuracy is quite high, high number of losses and profits are mispredicted.

Predicting Token Price Movement with Market Condition Data

To run the following models we created a database containing the price change of a particular token and the market conditions change in the same period. The price change is measured between two consecutive price entries from the same token. Hence, due to limited data, the time horizon of the price changes differ between and within tokens, but for each price change, the market conditions changes are computed using the same time horizon. This approach allows us to look at each entry as an independent self contained entry irrespective of its moment in time, in which we have information for the changes in the price and changes in the market conditions during the same time horizon. The dependent variable is called ‘Status’ and takes value 1 if the price of a token increases with respect to its last observation, and 0 otherwise [Figure 1 in Appendix E]. Unfortunately, given the market conditions on Cryptopunks, our dataset is strongly unbalanced. 83% of the entries experienced a rise in price, which can negatively affect the performance of our models described below.

Logistic Regression

As an alternative approach, we used a Logistic Regression Model to predict if the price of a particular Cryptopunk was going to increase (value of 1) or decrease (value of 0), based on changes in the market variables. Even when the model obtained an accuracy of 84.4% when implemented in the test data, the results are not promising at all. This accuracy is given the unbalance of the data, as the model predicted an increase in the price every single time. [See Table 1 in Appendix E].

Neural Networks

Continuing with our analysis, we decided to implement Neural Networks, specifically the Multi-Layer Perceptron Classifier Model, to predict whether the price of a particular Cryptopunk is going to rise or fall. We tried different models (one, two and three hidden layers) using the Relu activation function. Nevertheless, given the unbalanced nature of our data, we faced the same problem as the case of Logistic Regression, all our predictions tell us that the price is going to increase. So again, the accuracy of the model implemented in the test data is 84.4%. [See Table 1 in Appendix E].

Random Forest

Finally, we used a Random Forest Model to predict whether the price of a particular Cryptopunk is going to rise or fall, expecting to hopefully obtain better results than the Logistic Regression and Neural Network models. Using the Grid Search Approach we found that the optimal model for this case had 5 max features, 0.4 max samples and 100 base estimators in the ensemble. Our model obtained an accuracy of 87.2% in the test data, outperforming the Logistic Regression and Neural Networks models [Table 1 in Appendix E]. However, as the data is heavily unbalanced, this level of accuracy is not informative at all about the performance of the model. Hence, we decided to rely on other techniques to analyse and classify the Cryptopunks tokens.

CONCLUSION

Based on data we collected, we concluded that we were unable to provide a clear trading strategy for Cryptopunks. Our initial hypothesis that prices of Cryptopunks could be determined solely on the attributes they possess was wrong, as our analysis showed a weak relationship between the two. We tried utilizing the K-Means algorithm to group tokens with similar traits together, thereby forming clusters with different prices. Even though performance of the model seemed to show some improvement after some feature engineering, we were not confident that the results could provide a significant advantage for trading.

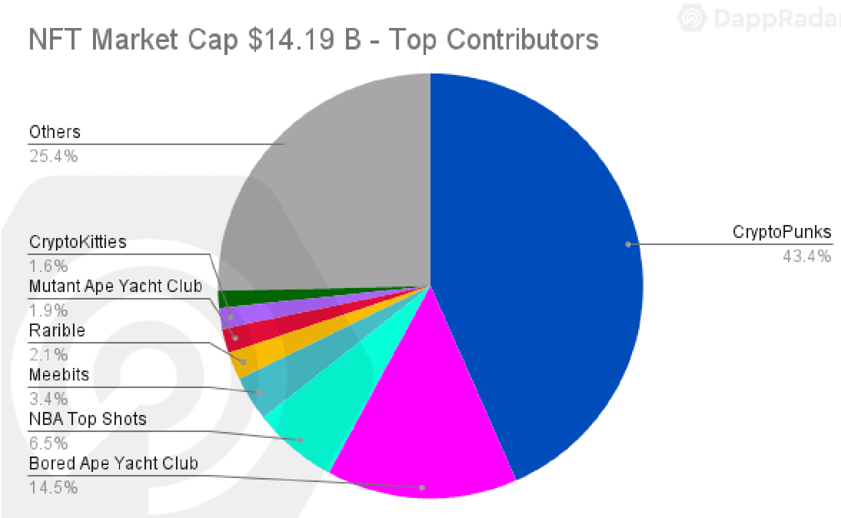
Instead of trying to predict the price of a single token, we decided to just predict the price movement of tokens. Unfortunately, the results again did not show any promise for adding any advantages for trading, even with the additional information about historical market conditions we had collected. All models suggested that the majority of the tokens would increase in price, which made sense as the NFT market is still new, with astronomical gains made in the last year. It had yet to fully develop and go through a bear market. It will be interesting to see which assets retain their value compared to others during that time.

As the market matures, more data can be collected and more research can be done on external factors worth including in our prediction models. Then, it will be interesting to backtest and develop various trading strategies across various assets in the NFT market.

APPENDIXES

APPENDIX A

Figure 1: NFT Market Capitalization Q3 2021

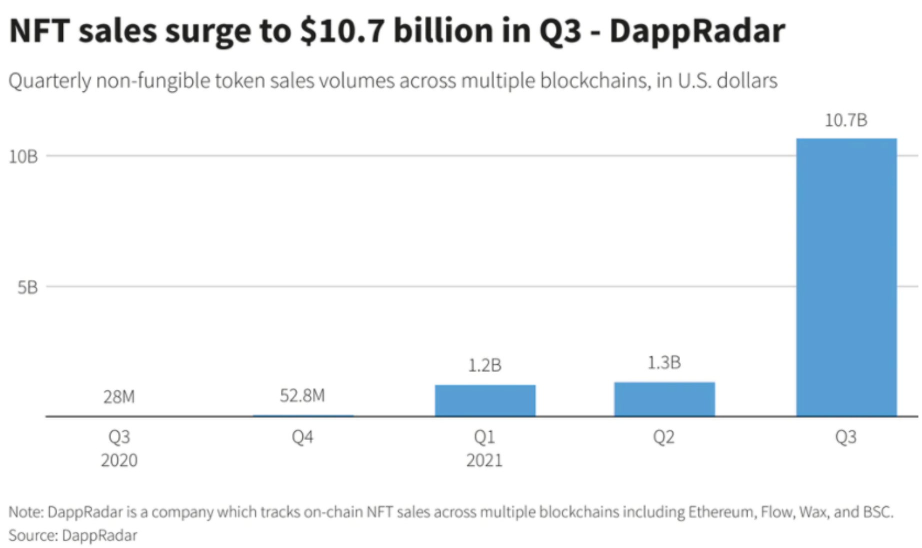


Source: Adapted [8]

Figure 2: Top NFT sales Q3 2021

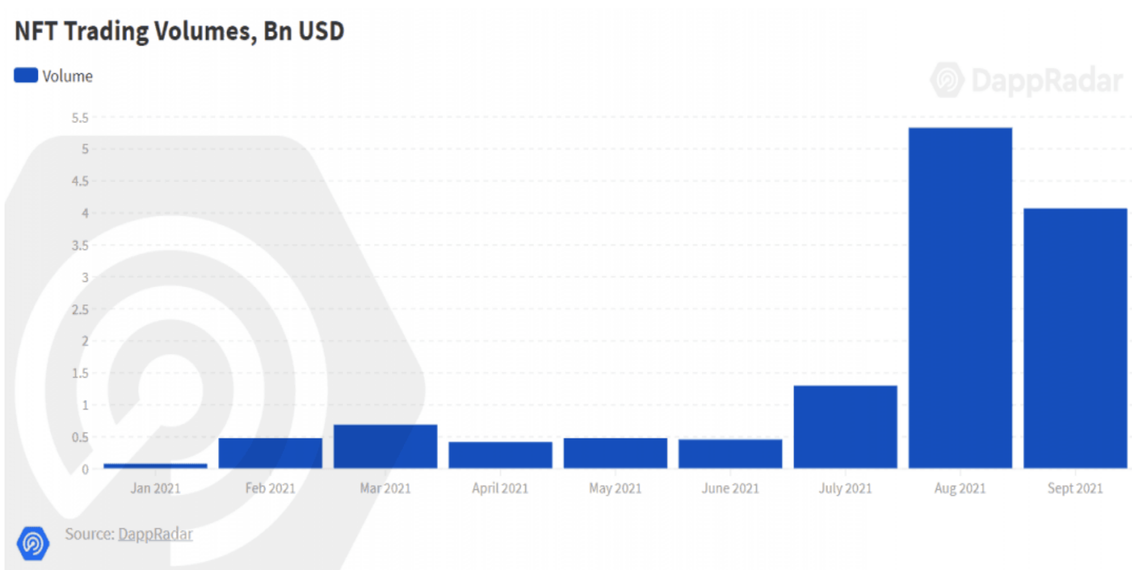
Source: Adapted [8]

Figure 3: NFT sales surge in Q3 2021



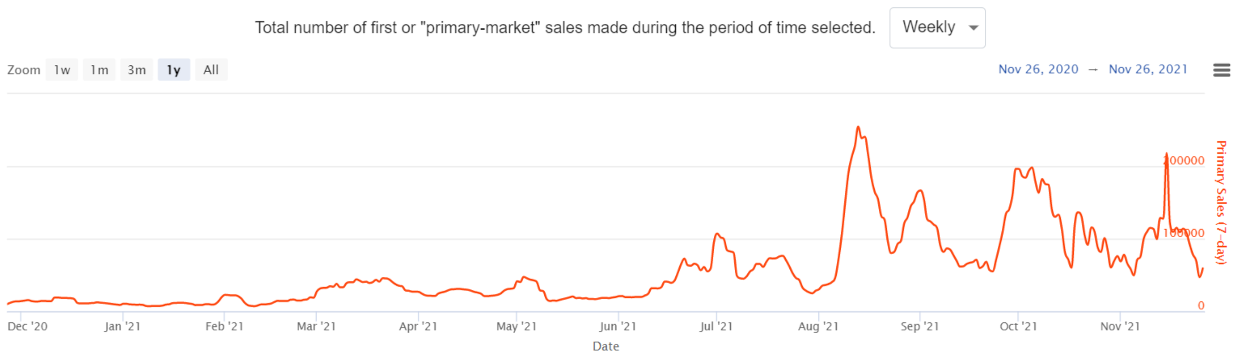
Source: Adapted [6]

Figure 4: NFT sales volumes through Sep 2021



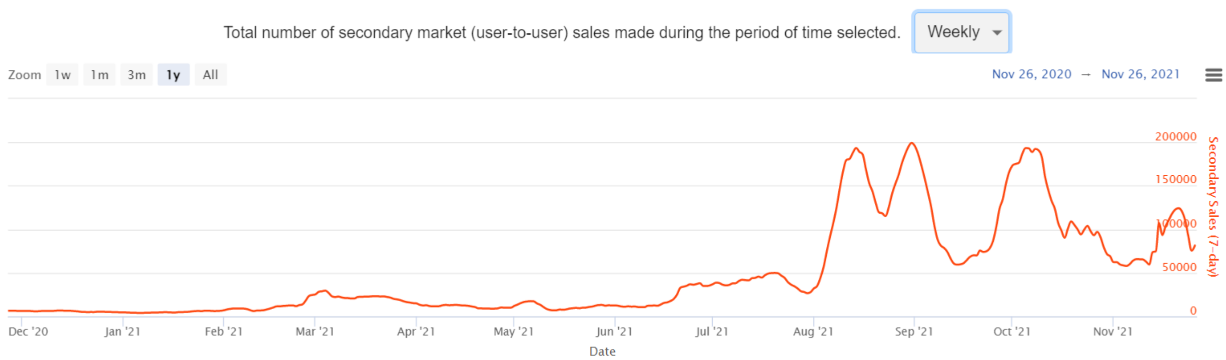
Source: Adapted [8]

Figure 5: Weekly primary sales of NFTs on the ETH blockchain (aggregated weekly)



Source: Adapted [10]

Figure 6: Weekly secondary sales of NFTs on the ETH blockchain (aggregated weekly)

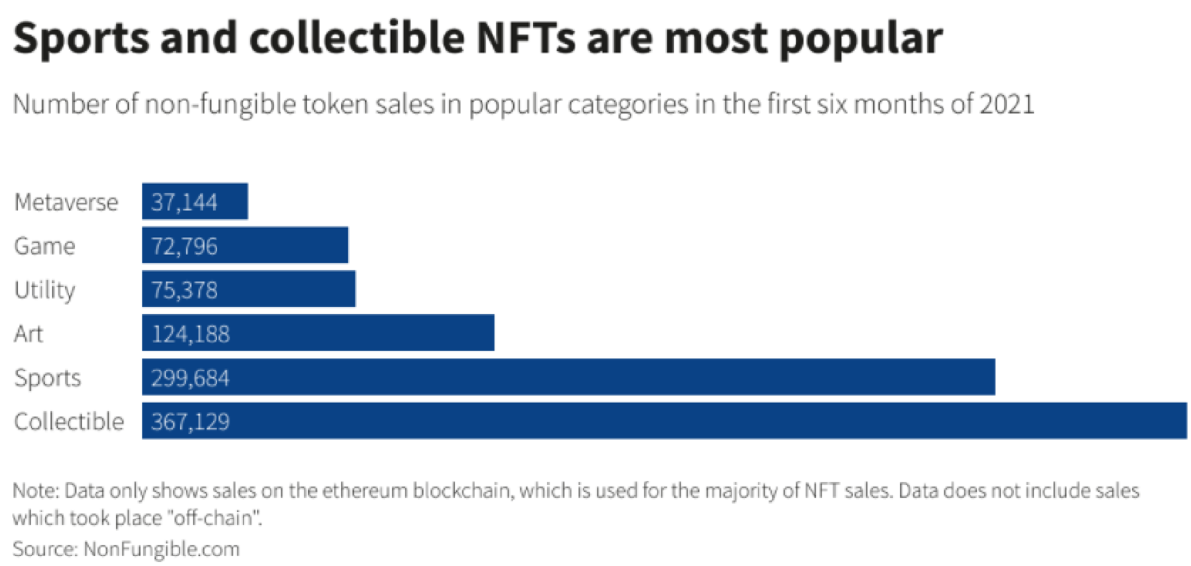


Source: Adapted [10]

Figure 7: Weekly money spent on NFTs on the ETH blockchain (aggregated weekly)

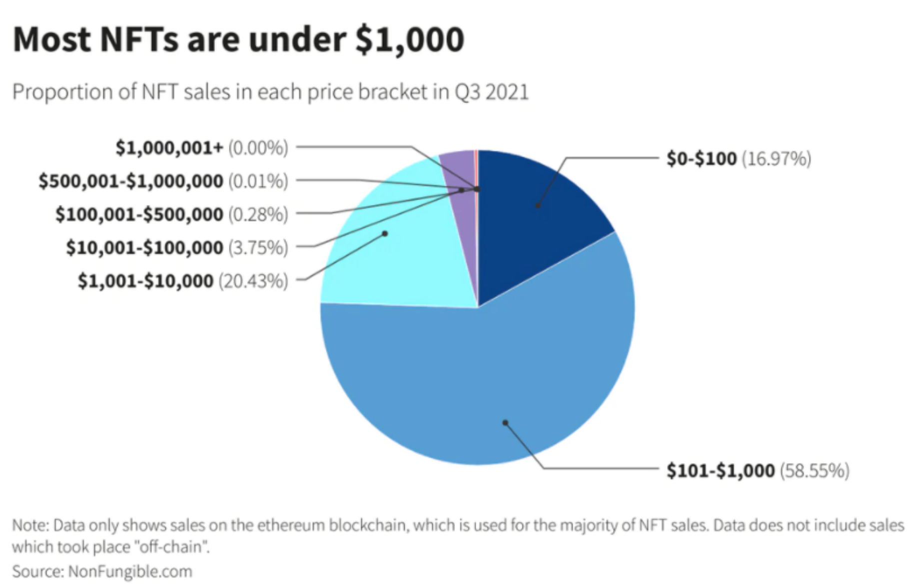
Source: Adapted [10]

Figure 8: NFT types



Source: Adapted [11]

Figure 9:NFT price ranges



Source: Adapted [6]

APPENDIX B

Figure 1: Types of CryptoPunks

	trait_count
trait_value	
Alien	9
Ape	24
Female	3840
Male	6039
Zombie	88

Figure 2: Attributes (excluding type) of CryptoPunks

	trait_count
trait_value	
Earring	2459
Cigarette	961
Hot Lipstick	696
Purple Lipstick	655
Mole	644
...	...
Orange Side	68
Tiara	55
Pilot Helmet	54
Choker	48
Beanie	44

87 rows × 1 columns

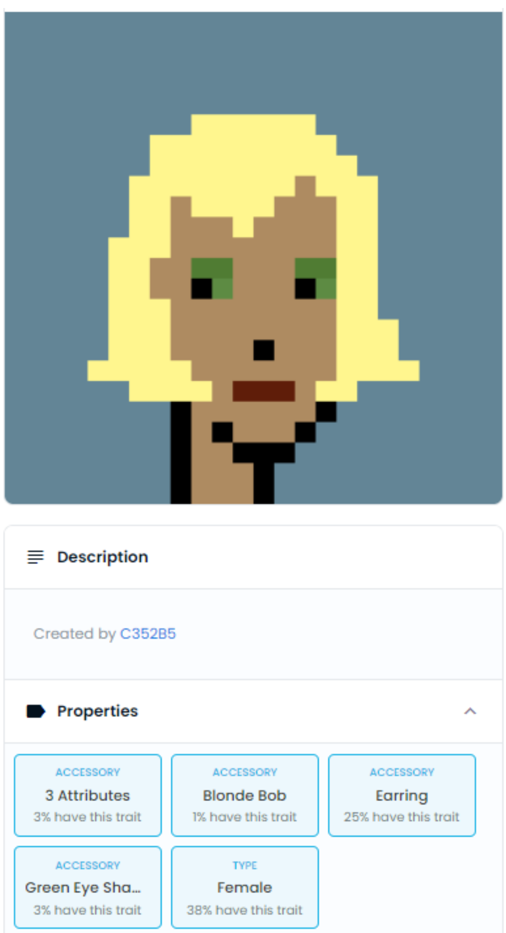
Figure 3: Number of CryptoPunks corresponding to different levels of attributes

	trait_count
trait_value	
0	8
1	333
2	3560
3	4501
4	1420
5	166
6	11
7	1

Figure 4: Snippet of Asset Data obtained through OpenSea API

id	token_id	trait_type	trait_value	trait_count
176533	0	type	Female	3840
176533	0	accessory	Green Eye Shadow	271
176533	0	accessory	Earring	2459
176533	0	accessory	Blonde Bob	147
158831	1	type	Male	6039

Figure 5: CryptoPunk #0 corresponding to first 4 rows of Asset Data in Figure 4



Source: Adapted [13]

Figure 6: Score calculation for CryptoPunk #5217 (total score = 648)

	id	token_id	trait_type	trait_value	trait_count	total_trait_count	rarity	score
5217	178333	5217	type	Ape	24	10000.0	0.002400	416.666667
23701	178333	5217	accessory	Gold Chain	169	27539.0	0.006137	162.952663
23702	178333	5217	accessory	Knitted Cap	419	27539.0	0.015215	65.725537
43966	178333	5217	num_attributes	2	3560	10000.0	0.356000	2.808989

Figure 7: CryptoPunk #5217 corresponding to Figure 6

Source: Adapted from [14]

Figure 8: Distribution of tokens scores (rarity)

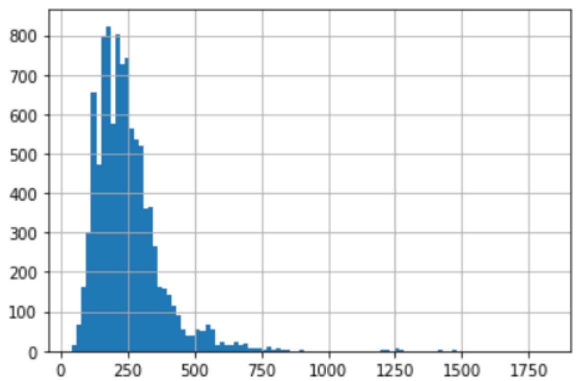


Figure 9: Score of attribute vs max transaction price of an attribute

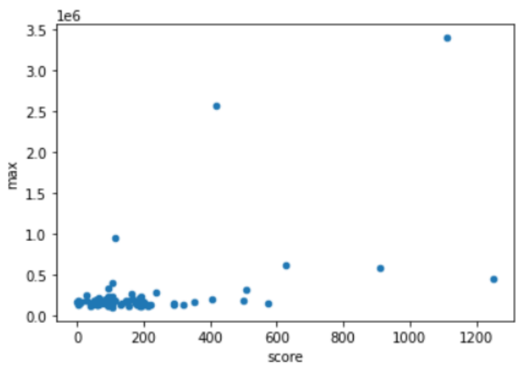


Figure 10: Score of token vs sum of average of the attribute prices that the token has

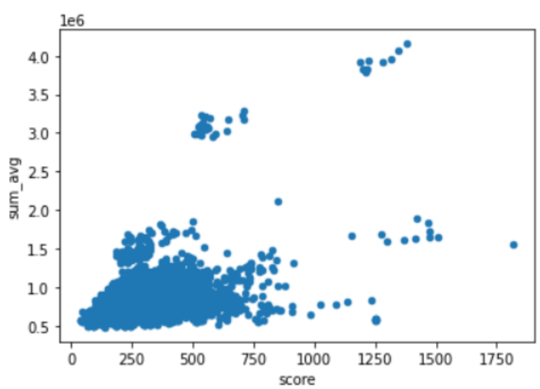


Figure 11: Twitter sentiment compared against ETH prices (limited to 910 data points due to time intensive data scraping)

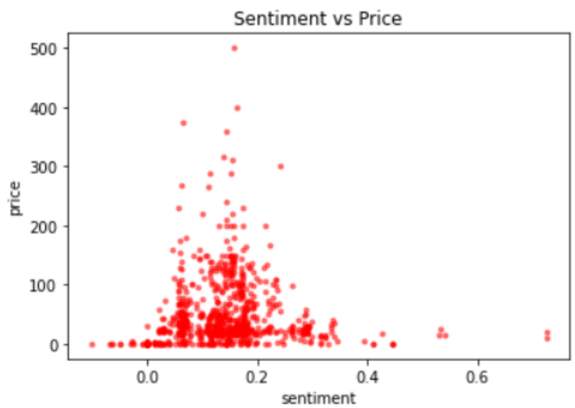


Figure 12: Max price of a token by score including outliers

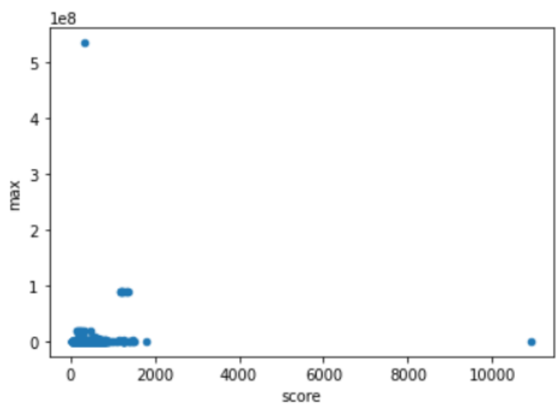
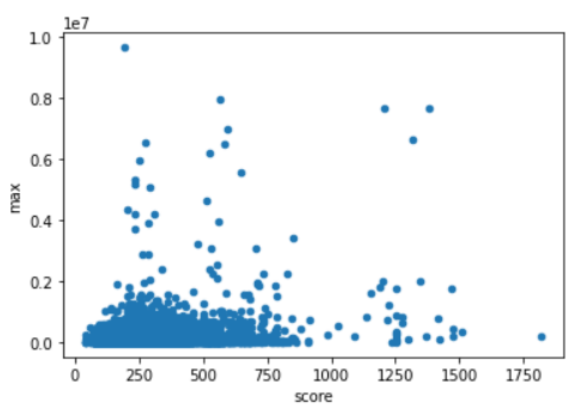


Figure 13: Max price of a token by score excluding outliers



APPENDIX C

Figure 1: Average price per cluster

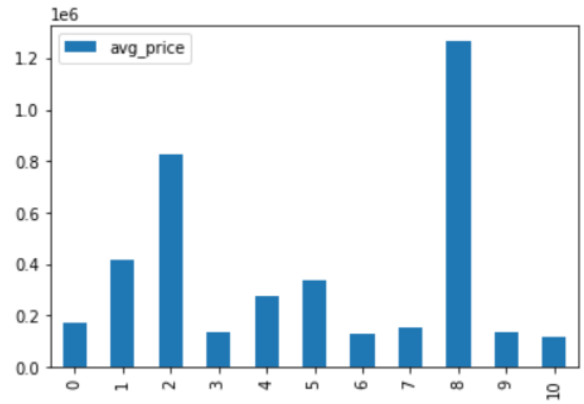
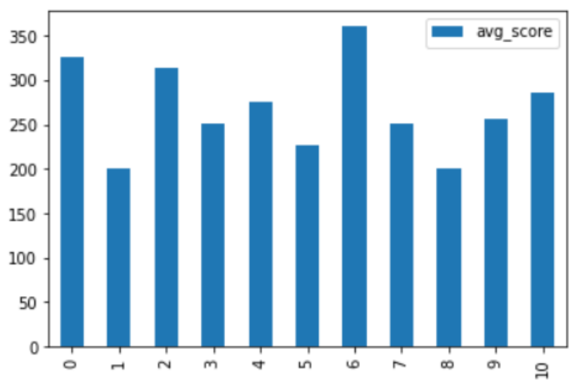


Figure 2: Average score per cluster



APPENDIX D

Figure 1: NFT Score replacing traditional one-hot encoding for each token & trait

trait_value token_id	3D Glasses	Alien	Ape	Bandana	Beanie	Big Beard	Big Shades	Black Lipstick	Blonde Bob	Blonde Short	...	Tiara	Top Hat	VR	Vampire Hair	Vape	Welding Goggles	Wild Blonde
0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	187.340136	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	51.474766	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 2: Principal Component Score (index is token_id and column is Principal Component)

	0	1	2	3	4	5	6	7	8	9	...	82	83	84	85
0	2.868173	0.015396	0.077127	0.535661	-0.335486	-0.322381	0.452947	-0.589473	0.104624	-0.681804	...	0.554771	0.696739	-0.513933	1.184192
1	-1.365979	0.277975	0.636110	0.359752	1.001173	0.704619	-0.480693	0.012484	-1.335782	1.398615	...	-1.242636	2.645720	-0.353439	-0.066309
2	1.197628	-0.102925	-0.634811	0.392736	1.543410	-0.065400	-0.379572	-0.437902	0.629324	-0.753663	...	-0.413878	0.375271	-0.670773	-0.128514
3	-0.968424	0.216092	-2.133291	-0.859155	0.301874	-1.588515	-0.773756	-0.414124	1.316726	-0.115280	...	-0.232928	-1.034752	0.961075	-0.017721
4	-1.473681	-0.152193	-1.082115	-1.106225	2.376299	-0.787320	0.089004	-1.688375	-0.995726	0.418189	...	-0.796541	1.175936	-1.033122	-0.379432

Figure 3: Cumulative Contribution

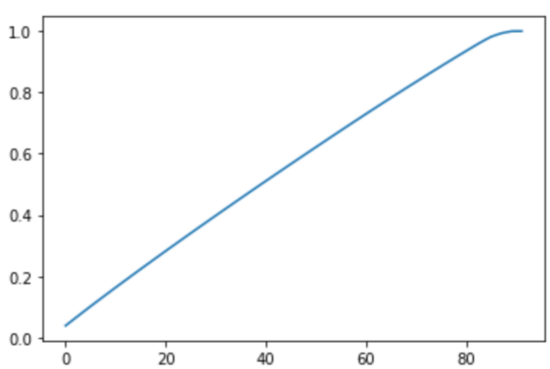


Figure 4: Plot of NFT scores (x-axis is 1st PCA and y-axis is 2nd PCA)

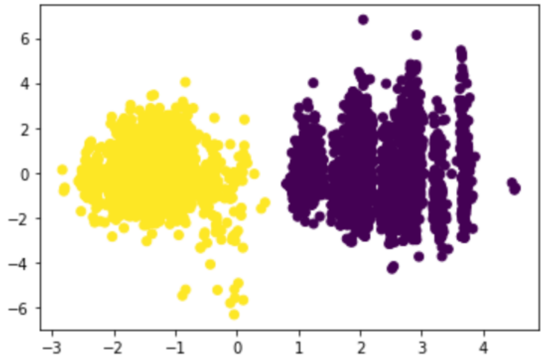
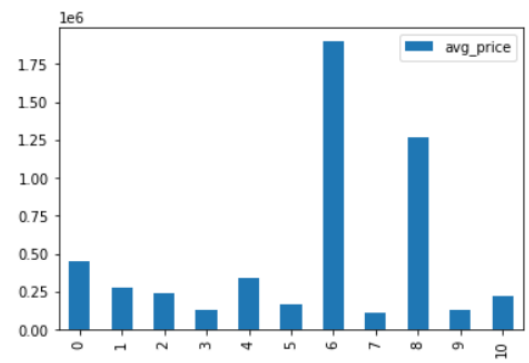


Figure 5: Average price and score of clusters (k=11)



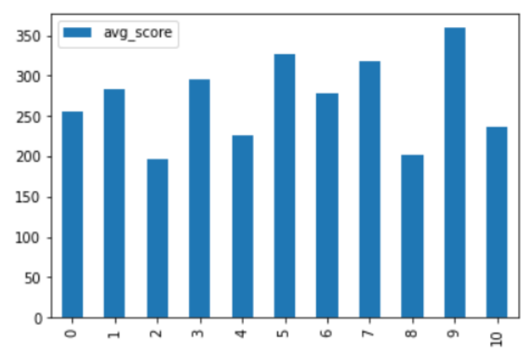


Figure 6: Cluster details with k=2 (left is 1st cluster and right is 2nd cluster)

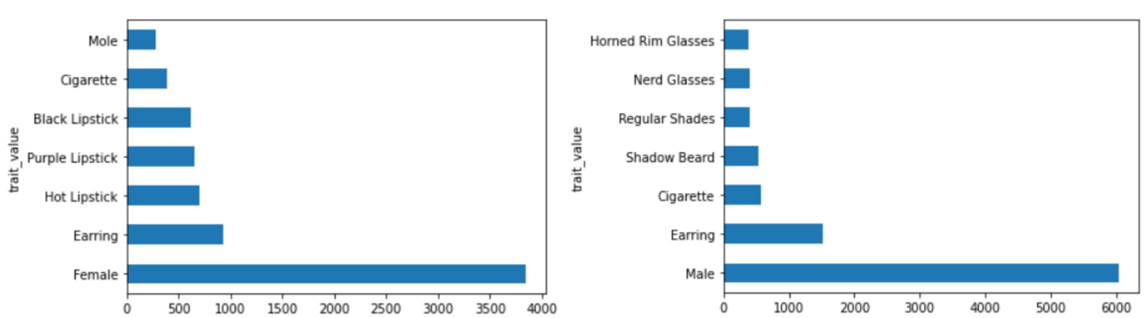
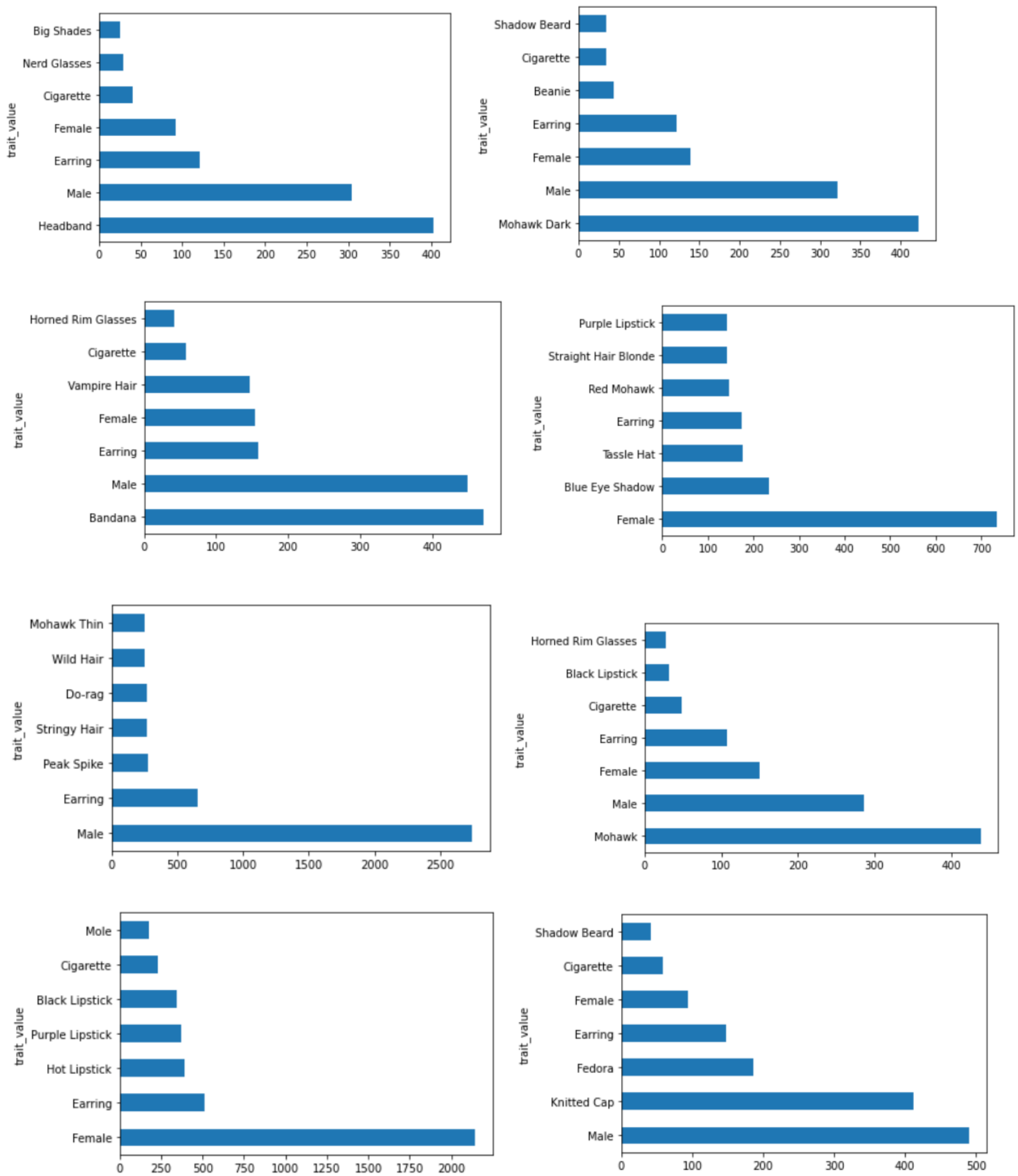
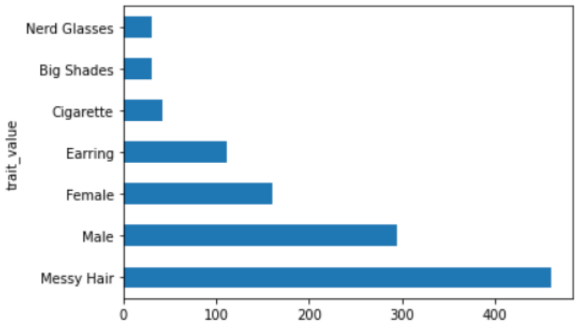
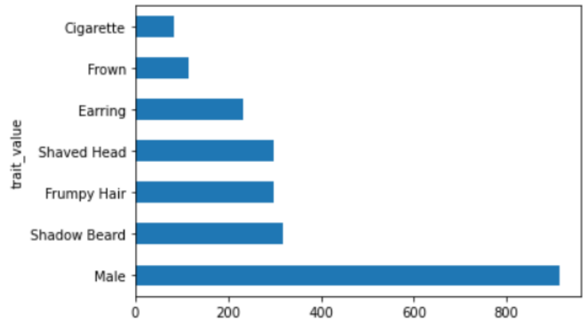


Figure 7: Cluster details with k=11 (clusters from left to right)




Nerd Glasses
Big Shades
Cigarette
Earring
Female
Male
Messy Hair
trait_value
0
100
200
300
400


Cigarette
Frown
Earring
Shaved Head
Frumpy Hair
Shadow Beard
Male
trait_value
0
200
400
600
800

APPENDIX E

Figure 1: Token database including market conditions by month

	token_id	Date	max_bid_amount_usd_price	max_Secondary_Sales_USD	max_Unique_Buyers	max_Unique_Sellers	max_Average_USD
1	0	2017-7	8.548649	18.146958	1.222222	1.666667	1.464618
2	0	2018-11	-0.115263	3.915613	7.425000	1.875000	0.027049
3	0	2019-4	1.318066	0.076179	0.169139	0.252174	-0.076069
4	0	2019-5	-0.306185	0.121964	0.081218	0.062500	0.012660
5	0	2019-12	0.282507	0.210784	0.171362	0.176471	-0.030508

… continue …

max_Number_of_Sales	max_Secondary_Sales	max_Active_Market_Wallets	max_Sales_USD	status
6.584906	6.584906	1.321429	18.146958	1.0
3.786070	3.786070	5.046154	3.915613	0.0
0.164761	0.164761	0.162850	0.076179	1.0
0.107988	0.107988	0.074398	0.121964	0.0
0.248892	0.248892	0.154786	0.210784	1.0

Table 1: Returns of alternative models

	Logistic Regression	Neural Networks	Random Forest
True Positive	1,959	1,959	1,887
True Negative	0	0	137
False Positive	362	362	225
False Negative	0	0	72
Accuracy	**0.844**	**0.844**	**0.872**

APPENDIX F

Figure 1: Confusion matrix for Logistic Regression without class weighting

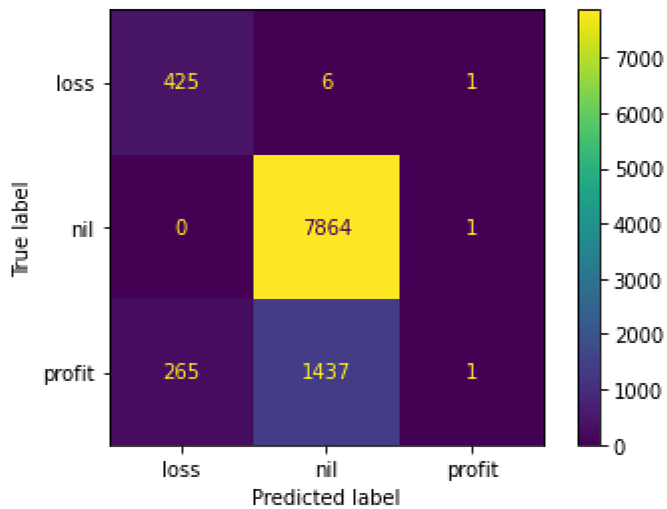


Figure 2: Confusion matrix for logistic regression with class weighting

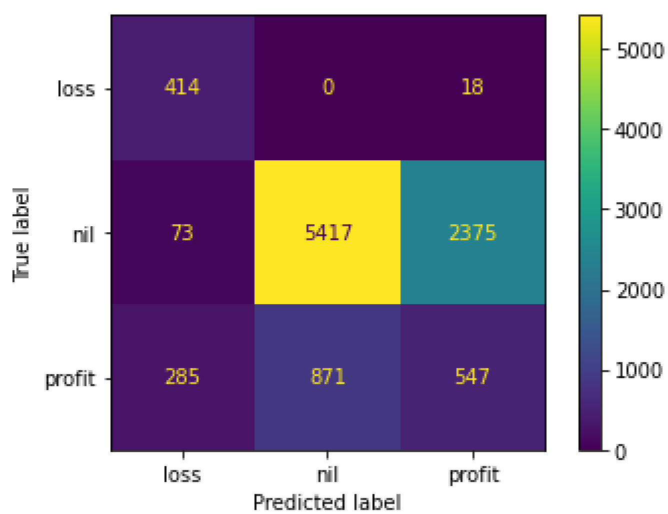


Figure 3: Confusion matrix for logistic regression with threshold change

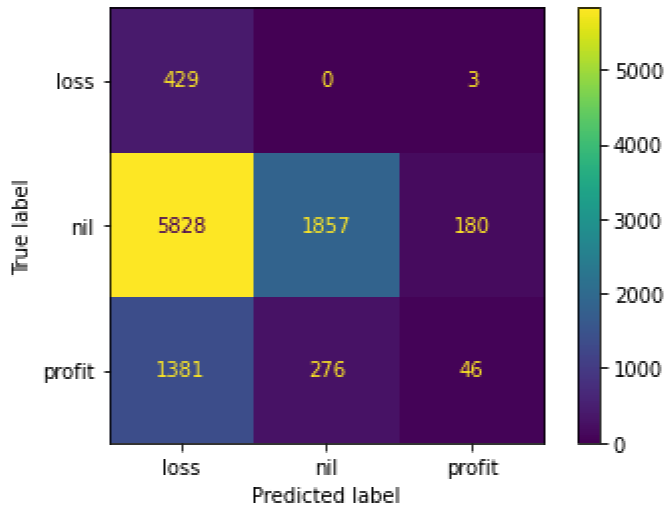


Figure 4: Confusion matrix for Gradient Boosted Classifier

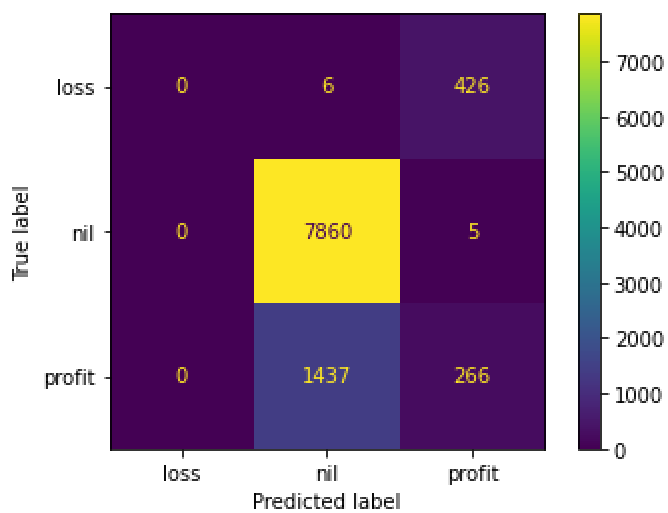


Figure 5: Format of data used for Classification

	type	num_attrs	attribute0	attribute1	attribute2	attribute3	attribute4	attribute5	attribute6
0	Female	3	Blonde Bob	Earring	Green Eye Shadow				
1	Male	2	Mohawk	Smile					
2	Female	1	Wild Hair						
3	Male	3	Nerd Glasses	Pipe	Wild Hair				
4	Male	4	Big Shades	Earring	Goat	Wild Hair			

REFERENCES

1. "Cryptopunk #8102 - cryptopunks | OpenSea," *OpenSea*, [Online]. Available: https://opensea.io/assets/0xb47e3cd837ddf8e4c57f05d70ab865de6e193bbb/8102. [Accessed: 07-Dec-2021].
2. "Cryptopunks," *Larva Labs*, [Online]. Available: https://www.larvalabs.com/cryptopunks. [Accessed: 29-Nov-2021].
3. S. Kaczynski and S. D. Kominers, "How nfts create value," *Harvard Business Review*, 19-Nov-2021. [Online]. Available: https://hbr.org/2021/11/how-nfts-create-value. [Accessed: 29-Nov-2021].
4. M. Clark, "NFTs, explained," *The Verge*, 03-Mar-2021. [Online]. Available: https://www.theverge.com/22310188/nft-explainer-what-is-blockchain-crypto-art-faq. [Accessed: 29-Nov-2021].
5. "ERC-721 non-fungible token standard," *ethereum.org*, 13-Sep-2021. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/. [Accessed: 29-Nov-2021].
6. E. Howcroft, "NFT sales surge to $10.7 bln in Q3 as Crypto Asset Frenzy hits New highs," *Reuters*, 04-Oct-2021. [Online]. Available: https://www.reuters.com/technology/nft-sales-surge-107-bln-q3-crypto-asset-frenzy-hits-new-highs-2021-10-04/. [Accessed: 29-Nov-2021].
7. "Non-fungible token (NFT) API overview," *OpenSea Developer Documentation*. [Online]. Available: https://docs.opensea.io/reference/api-overview. [Accessed: 01-Oct-2021].
8. P. Herrera, "Dapp Industry Report: Q3 2021 overview," *DappRadar Blog RSS, 07-Oct-2021*, [Online]. Available: https://dappradar.com/blog/dapp-industry-report-q3-2021-overview. [Accessed: 29-Nov-2021]
9. "Ethereum USD (ETH-USD) price, value, news & history," Yahoo! Finance, 07-Dec-2021. [Online]. Available: https://finance.yahoo.com/quote/ETH-USD?p=ETH-USD. [Accessed: 07-Dec-2021].
10. "Market history: NFT sales and Trends," *NonFungible.com*, 29-Nov-2021. [Online]. Available: https://nonfungible.com/market/history. [Accessed: 29-Nov-2021].
11. E. Howcroft, "NFT sales volume surges to $2.5 bln in 2021 First Half," *Reuters*, 06-Jul-2021. [Online]. Available: https://www.reuters.com/technology/nft-sales-volume-surges-25-bln-2021-first-half-2021-07-05/. [Accessed: 29-Nov-2021].

12. rarity.tools Rarity.tools, “Ranking rarity: Understanding rarity calculation methods,” *Medium*, 31-Oct-2021. [Online]. Available: https://raritytools.medium.com/ranking-rarity-understanding-rarity-calculation-methods-86ceaeb9b98c. [Accessed: 02-Dec-2021].
13. “Cryptopunks: Details for punk #0,” *CryptoPunks: Details for Punk #0,* [Online]. Available: https://www.larvalabs.com/cryptopunks/details/0. [Accessed: 29-Nov-2021].
14. “Cryptopunks: Details for punk #5217,” *CryptoPunks: Details for Punk #5217*, [Online]. Available: https://www.larvalabs.com/cryptopunks/details/5217. [Accessed: 29-Nov-2021]
15. “sklearn.linear_model.LogisticRegression — scikit-learn 1.0.1 documentation,” *sklearn.linear_model.LogisticRegression — scikit-learn 1.0.1 documentation*, [Online]. Available:https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed: 29-Nov-2021]
16. “sklearn.metrics.roc_auc_score — scikit-learn 1.0.1 documentation,” *sklearn.metrics.roc_auc_score — scikit-learn 1.0.1 documentation*, [Online]. Available:https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics. [Accessed: 29-Nov-2021]
17. “sklearn.ensemble.GradientBoostingClassifier — scikit-learn 1.0.1 documentation,” *sklearn.ensemble.GradientBoostingClassifier — scikit-learn 1.0.1 documentation*, [Online]. Available:https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. [Accessed: 29-Nov-2021]