Machine Learning-Based Prediction of CryptoPunk NFT Investment Profitability

Liuyang Xu, Hao Tang, Jinyu Hu

Messy Data and Machine Learning

May 5th, 2022

Abstract

NFT's has been a popular topic recently where people have started to consider digital art to be valuable. CryptoPunks are one of the earliest versions of NFTs and are generated using a program that combines different colors and attributes in random combinations along with a significant price.

This paper mainly focuses on the question of whether CryptoPunks will be sold at a price level that exceeds the average appreciation increase. The data was scraped from the Lava Labs. The four main branches of the features used in the prediction model are sale information, punk gender, punk image color composition, and punk attributes.

K Nearest Neighbor, Linear Regression, Logistic Regression, Random Forest, and One-hidden-layer Neural Network are used to predict the binary outcome separately, and the result showed the Random Forest model is the best model for prediction. The paper also explores the real sale price prediction using Random Forest, Neural Network, and Linear regression. The Random Forest model also provided the best prediction result. The model paper obtained can be used to provide investors with buying and selling recommendations.

Keywords: CryptoPunks, KNN, Regression, Random Forest, Neural Network

Contents

1. Introduction

Traditional database operation has a great centralized character, so the read and write authority is in the hands of a company or a centralized organization. Blockchain, on the other hand, is a globally distributed database storage system that can operate collaboratively.

Anyone who can set up server nodes can participate in the blockchain, and all blockchain nodes around the world are connected. Any data read and written will be synchronized in the same way. This list of records with identical data in all nodes worldwide is cryptographically linked together and growing. It can be used to record transactions and track assets. It is used in almost all areas of life today, including logistics, healthcare, and even artwork.

The ability to record assets has brought about the derivative development of blockchain, and among them Non-Fungible Tokens, representing ownership of unique items. This technology allows us to tokenize things like art, collectibles, and even real estate. They can only have one official owner simultaneously and are guaranteed by the blockchain that no one can modify the record of ownership or copy/paste a new NFT into existence.

In a way, the art auction shoot is essentially scarcity, and digital artwork is difficult to speak of scarcity because of the ease of replication and dissemination. NFT’s are a good solution to this problem: indivisible and inimitable, fidelity, and indicate the right of attribution. These characteristics are a guarantee of uniqueness.

On March 11, 2020, Christie's sold an NFT digital artwork *Everydays: The Frist 5000 Days*, for $69 million. It was the third highest-priced work ever sold at Christie's, and it also placed the painting's author, American graphic designer Beeple, among the three most expensive living artists. He did not sell a single work until 2020, after which his work has rivaled Picasso's prices.

People have started to consider digital art to be precious. CryptoPunks are the earliest versions of NFTs and are generated using a program that combines different colors and attributes in random combinations, but it also has a significant price. This is very different from traditional artwork, where the cost of an image is judged by the fame of the artist, the use of materials, the idea and creativity, and the historical accumulation. So, this leads to the question of how to predict the price of a CryptoPunk picture.

We intend to use various types of information and transaction records of CryptoPunk pictures to build a price prediction model using machine learning methods to explore the price changes of CryptoPunk pictures and to provide investors with buying and selling recommendations.

2. Data Collection

Larva Labs provides the description of the CryptoPunks, as well as the png file of the picture, the attributes the image has, and the sale history(price, owner, buyer, transaction date, transaction type) of each punk. We planned to use the color information, attributes, CryptoPunk gender, Current Market Status, sale history, etc., as variables to predict the price of the CryptoPunk.

All these pieces of information are likely to influence the transaction price of punk images, so we scraped all of these data down from the website. Since the transaction information is constantly updated, we selected all 21,016 transactions in April 2022 and before as our dataset.

By default, each page of transaction records displays 96 punks, which is very tedious to scrape. Therefore, we modified the number of punks displayed on a single page to the number of punks that were sold so that we could directly obtain the links to all punk transaction history pages at once. In each link, we can find the above-mentioned image, sale history, attributes, and gender information.

To avoid the HTTP error 429 caused by too many frequent visits to the web page, the scraping process was set to visit the website every three seconds, and the total scraping time took about 7 hours.

After downloading all the punk images scraped from the web page, we used the PNG package to store them in RGBA format with 24×24 pixels to ensure that this type of data information can accurately reflect the image content of the punk. The image information can also be restored as the original picture in this way.

3. Date Cleaning

After the web scraping, we got four lists composed of multiple data frames, and each of the lists has 6674 independent Punk information.

3.1. Sale history dataset

The Punk Sale History contains transaction information of the corresponding Punk, and it is mainly divided into five categories: “Claimed”, “Offered”, “Sold”, “Bid”, “Offer Withdrawn”, “Bid Withdrawn”, and “Transfer”. These data frames contain multiple times of records in the punk, such as “Bids”, “Offered” etc. Moreover, each category has its own corresponding transaction time, transaction code, and transaction amount (Bitcoin and USD).

Not all punks contain the complete set of these five categories. For example, some punks lack “Offered” and some lack “Bid”, but all punk images should have “Sold” and “Claimed”.

(1) Dollar cleaning and Date cleaning

In the Sale History, we first cleaned up the Price column and extracted the “Dollar” from it because it includes two different price types, Bitcoin and USD, and we need USD as a reference in the Sale Data. The further cleaning on the “dollar” column ensured it had no symbol or character strings(just numeric and with the same type of format). Then we cleaned up the date column. The original date expression is written as “Apr 20, 2022”, but we changed it to the “M-Day-Year” in numeric format. For example, “04-20-2022”, such format allowed us to read and extract the date information more easily in the following steps

(2) Transaction Type Cleaning

By observing the Punk Sale History, we found “Sold" and "Bid” have two different expressions. Sold is represented by "Sold" and "Sold *", and Bid is represented by "Bid" and "Bid*". After returning to the NFT website to do some analysis and comparisons, we found that there is no difference between the two types of expressions. Therefore, we unify the two forms as "Sold" and "Bid" respectively.

3.2. Punk gender

The Punk Gender is the "sex attribute" of punk characters. There are five categories in total, namely Female, Male, Zombie, Alien, and Ape.

We extract the gender information from the original text data.

figure1

3.3. Punk attribute

The Punk Attributes is an in-depth representation of Punk Gender. For example, the clothes worn by the punk characters, the color of their hair, and some objects that appear in the pictures, etc. Each individual punk will have its own properties but may share some common properties with other punks. There are a total of 86 attributes in Punk Attributes.

There are 6674 rows in the Punk Attributes, and each punk has at least two corresponding attributes. We want to pass each punk's transaction record to its corresponding punk's attribute. Firstly, we converted the list of datasets into an empty data frame format which contains 86 independent columns through 86 different attributes. If the punk attribute appears in its corresponding punk transaction, we marked it as 1, otherwise 0.

figure2

3.4 RGBA

RGBA is a model of color space, consisting of RGB color space and Alpha channel. RGBA stands for Red, Green, Blue, and Alpha channel (opacity parameter of the image).

In our collected dataset, the RGBA information of each punk image is stored in four 24×24 matrices, and each cell represents the color or transparency information of the corresponding pixel point. The RGBA data didn't require any cleaning.

4. Feature Engineering

4.1. Research Outcome Variable Generating

We consider each sale (presented as "sold") of each punk as an observation. The observation would contain the basic sale history information and other features that would affect the prediction.

People who invested or try to invest in NFTs nowadays are seeking significant profits. Whether the investment in CryptoPunk pictures can bring them returns beyond those brought by conventional investments such as stocks and futures has become a central concern for them. Based on the purpose of exploring the price changes of CryptoPunk pictures and to provide investors with buying and selling recommendations, we chose to create a new binary variable — "Over Ave Ratio" and specify the research question as "whether the punk picture with the features we select would be sold at a price level that exceeds the average appreciation increase". More specifically, "Over Ave Ratio" is a calculated by quotient (Sold Price/Last Sold Price) of each row whether larger than the 5 percent of the average quotient (Sold Price/Last Sold Price) of the same given year, larger marked as "1" and smaller marked as "0".

The formula for finding the threshold for each year:

$$Threshold = Ave[sold\ price(i|year)/last\ sold\ price(i|year)] * 0.05$$

year = 2017, 2018, 2019, 2020, 2021, 2022

Year	2017	2018	2019	2020	2021	2022
Threshold	1.0226	0.7504	0.1942	6.2050	454.0443	780.9025
Ratio(1|0)	1153 : 133	815 : 84	1068:0	1337:2993	2251 : 10090	237 : 1186

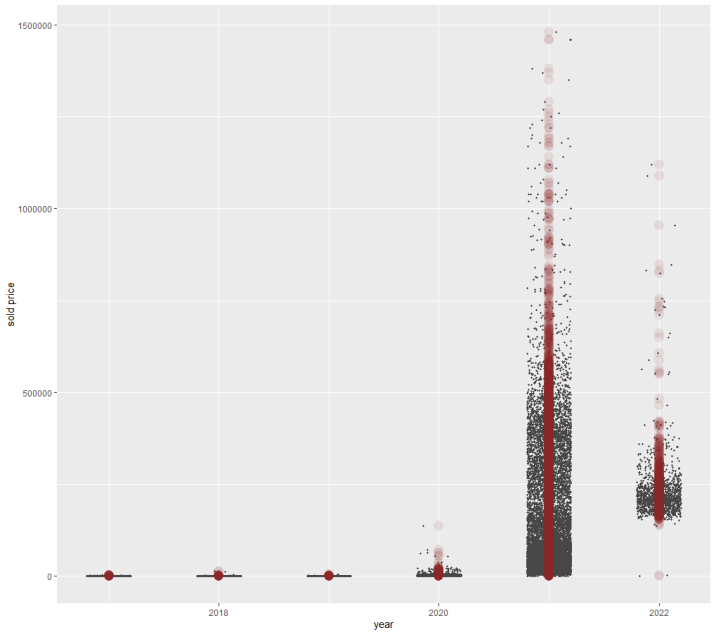


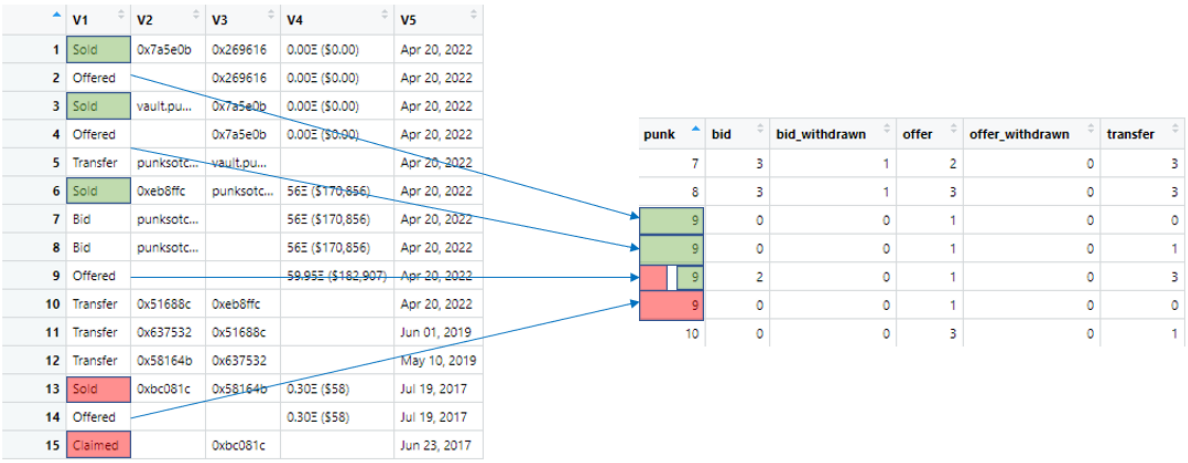
figure3

The figure above presents the general view of sold price information in each year. The opacity of the red shows the density of the price around that value. The reason we didn't use the average threshold for all year is that the market trend is constantly changing, thus the threshold in 2017 can't use directly to assess the data in 2019. For example, the threshold for 2017 is around 10, but the threshold for 2022 is near 8000. Therefore, we decided to calculate our average ratios separately and make an assessment inside of each year.

4.2. Research Observations Generating

We converted datasets of 6674 punk sale history into one data frame, which contains all 21347 punk transaction records. One of the key problems is the number of transaction records in each independent punk sale history.

Using each "Sold" and “Claimed” to distinguish each transaction record is the key to the conversion. For example, if there are five “Sold” in one punk, this punk will generate five transaction records, and each transaction record will contain some information on sale history, such as the number of “Bid” and “Offered”, the time and price of “Sold”, and the price of the largest “Bid” and “Offered” and many more.



	V1	V2	V3	V4	V5
1	Sold	0x7a5e0b	0x269616	0.00Ξ ($0.00)	Apr 20, 2022
2	Offered		0x269616	0.00Ξ ($0.00)	Apr 20, 2022
3	Sold	vault.pu...	0x7a5e0b	0.00Ξ ($0.00)	Apr 20, 2022
4	Offered		0x7a5e0b	0.00Ξ ($0.00)	Apr 20, 2022
5	Transfer	punksotc...	vault.pu...		Apr 20, 2022
6	Sold	0xeb8ffc	punksotc...	56Ξ ($170,856)	Apr 20, 2022
7	Bid	punksotc...		56Ξ ($170,856)	Apr 20, 2022
8	Bid	punksotc...		56Ξ ($170,856)	Apr 20, 2022
9	Offered			59.95Ξ ($182,907)	Apr 20, 2022
10	Transfer	0x51688c	0xeb8ffc		Apr 20, 2022
11	Transfer	0x637532	0x51688c		Jun 01, 2019
12	Transfer	0x58164b	0x637532		May 10, 2019
13	Sold	0xbc081c	0x58164b	0.30Ξ ($58)	Jul 19, 2017
14	Offered			0.30Ξ ($58)	Jul 19, 2017
15	Claimed		0xbc081c		Jun 23, 2017

	punk	bid	bid_withdrawn	offer	offer_withdrawn	transfer
7	3	1	2	0	3	
8	3	1	3	0	3	
9	0	0	1	0	0	
9	0	0	1	0	1	
9	2	0	1	0	3	
9	0	0	1	0	0	
10	0	0	3	0	1	

(punk_Sale_History[[11]]) (punk_Sale_History[[11]] after converted)

figure4

As we can see from the figure, the order of the V1 column is arranged by descending order of date. Therefore, the "Claimed" marked in green is the time when Punk was generated and went on sale (all "Claimed" dates are Jun 23, 2017). The first "Sold" marked in red in the figure is the

first mark of the transaction record of the Punk, so "Claimed" to the first red "Sold"(inclusive) is regarded as the first transaction record. And from the first red "Sold" to the first green "Sold" (inclusive) is regarded as the second transaction record, and the first green "Sold" to the second green "Sold" (inclusive) is regarded as the third transaction record, and similarly, the second green "Sold" to the third green "Sold" (inclusive) is regarded as the fourth transaction record.

4.3. Other Feature Generating

(1) Adding Last Sold Price

Before stepping into this one, the conversion on the transaction record had to be done, and then we added last_sold_price corresponding to the sold price, because this feature might be important for the prediction and treat it as a reference for the later fitted model process.

(2) Adding Gas Price

The Ethereum (ETH) gas prices are fees that are paid to miners whenever a payment transaction is initiated on the blockchain. This fee more than tripled between the certain period . From the line chart below, we can observe these gas prices were very low until 2020, after that, gas prices had a huge increase which might have been triggered by the Ethereum network starting to cope with increasing amounts as well as more complex transactions (*Raynor de Best, 2020*).

The first transaction record in each punk has no corresponding previous transaction record. To make our dataset more diverse without discarding the rows with NA. Instead of deleting rows, we used the average gas prices of all punks at the fixed period from Statista and corresponded it to the same time that the first transaction record occurred.

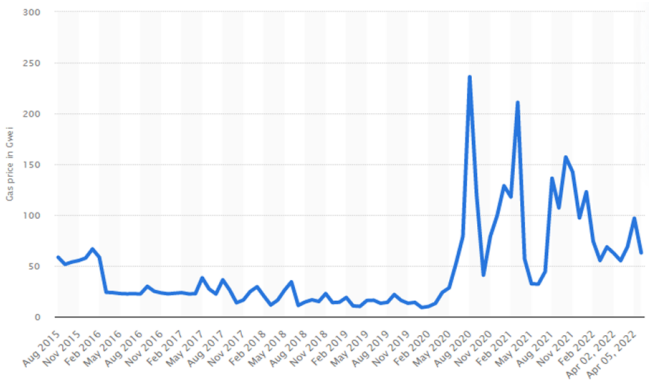


figure5 Average daily gas price of Ethereum from August 2015 to April 6, 2022

punk	sold_price	sold_date	last_sold_price
1	177275.00	2022-04-20	55.00
1	55.00	2017-08-12	72.00
1	72.00	2017-07-20	58.00
1	58.00	2017-07-19	NA
2	181899.00	2022-04-20	180524.00
2	180524.00	2022-03-09	76034.00
2	76034.00	2021-05-03	44201.00
2	44201.00	2021-03-13	41984.00
2	41984.00	2021-03-11	NA
3	212730.00	2022-04-20	186497.00
3	186497.00	2022-03-18	171379.00
3	171379.00	2022-03-10	172913.00
3	172913.00	2022-03-09	410275.00
3	410275.00	2021-10-02	40364.00
3	40364.00	2021-07-16	45760.00
3	45760.00	2021-02-23	NA

punk	sold_price	sold_date	last_sold_price
1	177275.00	2022-04-20	55.000
1	55.00	2017-08-12	72.000
1	72.00	2017-07-20	58.000
1	58.00	2017-07-19	22.310
2	181899.00	2022-04-20	180524.000
2	180524.00	2022-03-09	76034.000
2	76034.00	2021-05-03	44201.000
2	44201.00	2021-03-13	41984.000
2	41984.00	2021-03-11	210.890
3	212730.00	2022-04-20	186497.000
3	186497.00	2022-03-18	171379.000
3	171379.00	2022-03-10	172913.000
3	172913.00	2022-03-09	410275.000
3	410275.00	2021-10-02	40364.000
3	40364.00	2021-07-16	45760.000
3	45760.00	2021-02-23	117.850

(Before Filling—Punk1&Punk2&Punk3) figure6 (After Filling—Punk1&Punk2&Punk3)

As shown in the figure above, we extracted gas prices from Jan 2017 to April 2022 to replace NAs.

(3) RGBA Features Generating

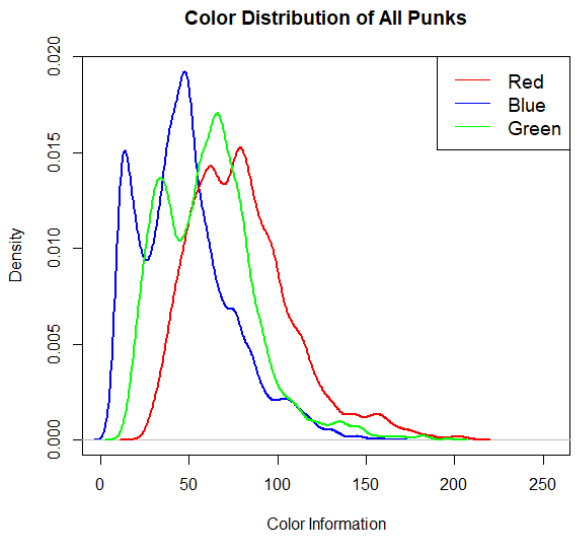
In our collected dataset, there are 2304 color values for each punk so using every pixel's color information is unrealistic for our model. The next step is to generate features that can reflect the color information without losing too much information while reducing the number of these features.

Considering that the color composition of the images may affect people's desire to buy, and RGBA reflects the various color information of the images respectively, we use the following formula to generate the model using the three color features and opacity feature

$$Total\ Color_k = \sum_{i,j=1}^{24} pixel\ color_{ij}\ , k = Red, Green, Blue, \alpha$$

where i and j represent the x and y coordinates of the pixel location.

The range of the value for the RGB feature we generated from the formula is [0,255]. The larger the value, the darker the corresponding color in that image. The density distribution of RGB and transparency density distribution of all Punks are shown in the following figures, respectively.



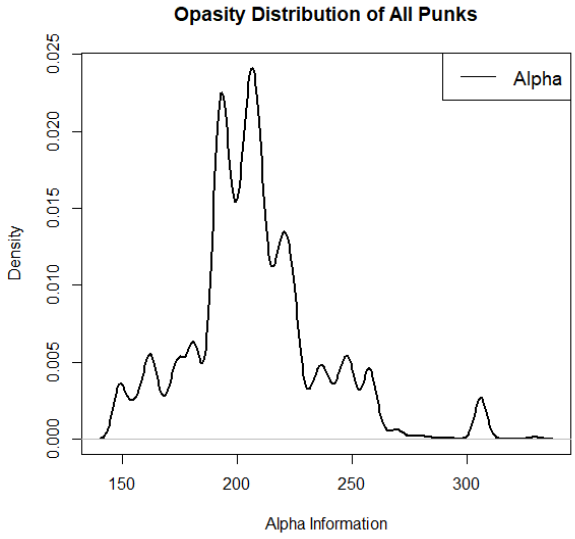


Figure7

Figure8

After combining the sale history with the other features, each transaction record will be added more information, such as the gender of the punk character in the transaction, the specific attribute, etc. Most of the columns are replaced by 86 Punk attributes, and each Punk Attribute appears as an independent binary variable in the Sale Data. The Sale Data should have 20,000+ rows and 100 columns, and each row represents an independent transaction record. The transaction record is defined by two consecutive “sold prices” in each punk.

5. Model

We shuffled the Final Sale Data dataset and divided it into three different datasets, train (50%), validate(25%,) and test(25%) based on the cross-validation method.

5.1. Modeling

5.1.1. K-Nearest Neighbour model

Supervised learning — KNN (K-Nearest Neighbor) is one of the Classification Methods used to predict the correct class for the test data by measuring the distance between data points from the test and all the training data points. The symbol "K" represents the number of nearest neighbors to new unknown variables that need to be predicted or classified. The logic behind this model is to create a fixed distance metric, determine the value of "K", given a new observation x(test), and find the K points in train data that are close to x. In order to calculate the distance between two points (inside of train data and train to the test data), the Euclidean distance was applied inside of the KNN package. The Euclidean distance' formula is

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

The smaller the K value is, the models tend to become more specific but might lose some generalizability by adding more noise at the same time. On the contrary, the relatively large value of K could make the model become too broad and fail to predict the data in both train and test accurately. Therefore, we set K values by decreasing order from 99 to 1 and each time the value decreases by 3 to take care of both concerns. Then we fit the KNN model on both train and test and compute the accuracy for the K value. After comparing the accuracy for different values of K, we choose the highest one from these values, which is K = 24.

5.1.2. Linear regression model

In this part, we fit a linear regression model. Simply, we can say that a linear regression model describes the relationship between a dependent variable, y, and one or more independent variables, X. The function below shows the linear regression model

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i, \quad i = 1, \cdots, n$$

The dependent variable is also called the response variable. Independent variables are also called explanatory or predictor variables. Continuous predictor variables are also called covariates, and categorical predictor variables are also called factors. In this part, because we have lots of independent variables, we are doing a multiple linear regression model. The multiple linear regression model interprets regression coefficients as comparisons between groups of units that differ in one predictor, with other predictors held fixed.

In this project, to predict the price of NFT change from 2017 to 2022, we first use the training part of the data to do the multiple linear regression model. We use the over average ratio as the dependent variable, y. If it is 1, the rest of the others are independent variables. There are 99 independent variables. The independent variables include the bid information, the offer information, the last sold price, gender difference, and 86 different attributes. The overlook of the model is presented below. (We didn't show all the variables in the sample model above because there are 99 variables)

$$OverAverageRatio_i = 1.48 + 0.05Bid + (-0.01)offer + (-0.39)gendermale + 0.1Beanie + \cdots + 0.29\, year_{2022} + \varepsilon_i$$

After getting the regression model, we found out that there are too many independent variables, and we think some of the variables may not be important enough or have a big influence on the result. This means that after we got this regression, directly using the linear regression may not be accurate, so we decided to use the AIC as the evaluation method to choose the best formula from the linear regression. The theory of choosing the best fit model first removes features from the full model, after that, start with $OverAverageRatioi = intercepts$ and evaluate the regression model. In the following step, change the independent variable already in the data to evaluate a different linear regression model. In this regression, we should choose between models with different numbers of predictors, so we use the package MASS and code stepAIC, which can easily help me to figure out the number of each model AIC. We put the direction =

"both" and after that, we got the different AIC numbers when the independent variable changed in the linear regression model. The function provides us the best regression model as

```
Call:
lm(formula = over_ave_ratio ~ bid + offer + offer_withdrawn +
    transfer + last_sold_price + gender + blue + alpha + Bandana +
    Beanie + Choker + Tiara + Orange_Side + Pigtails + Top_Hat +
    Rosy_Cheeks + Wild_White_Hair + Cowboy_Hat + Wild_Blonde +
    Red_Mohawk + Half_Shaved + Blonde_Bob + Vampire_Hair + Clown_Hair_Green +
    Straight_Hair + Dark_Hair + Purple_Hair + Gold_Chain + Tassle_Hat +
    Fedora + Clown_Nose + Hoodie + D_Glasses + Luxurious_Beard +
    Do_rag + Shaved_Head + Peak_Spike + Pipe + VR + Cap + Small_Shades +
    Clown_Eyes_Blue + Headband + Crazy_Hair + Mohawk + Frumpy_Hair +
    Wild_Hair + Messy_Hair + Stringy_Hair + Mole + Cigarette +
    year + month, data = train)
```

Finally, from the new formula AIC choice, we find out that the formula only exists 53 independent variables now. And this is the best fitting multiple linear regression model choices from the stepAIC method.

5.1.3. Logistic regression model

Supervised learning - logistic regression is one of the Classification Methods and has been widely used to calculate or predict the probability of binary events occurring. Compared to the Linear model where the outcome is continuous and can be any given value, the logistic model predicted the binary one, which is also a reason for setting the "Over Ave Ratio" in Sale Data. The formula for Logistic Function:

$$logit(p) = logit(\varepsilon[Y_i|x_{1,i\ldots,}x_i] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon_i$$

where p is the probability of the presence of the characteristic of interest.

(1) Standardization

By observing continuous variables in our dataset, we can see the range of price is large, and it's hard to see what distribution it might follow. Thus, standardizing makes it easier to compare scores, although those scores were measured on different scales. It also makes it easier to read results from regression analysis and ensures that all variables contribute to a scale when added together. Most importantly, it eases the interpretation of regression results.

(2) Choosing the predictors

The Logistic model is constructed by a response variable and 86 predictive variables, and the predictor contains 10 continuous variables, such as “Bid”, “Last Sold Price”, “Gender”, and “Red”, etc., and 76 categorical variables that mostly composed by Punk Attributes, such as “Bandana”, “Big Beard”, “Top Hat”, etc. Before fitting the logistic regression, we checked the correlation between all variables and found some variables had no correlation (full of NA) with any others in the dataset. Although these variables have almost zero effect on the results of the fitted model, deleting them allows us to better observe the results.

(3) Result interpretation

After fitting the model from the train to the validate dataset, we had the list of predictive probabilities. The AUC score we computed for the logistic model’s precision is 0.87, which suggests the model has good discriminatory ability 87% of the time, and the model will correctly assign the higher probability to the transaction record with corresponding predictors.

5.1.4. Random Forest model

In this part, we use the random forest model to predict. Random forest models are developed by decision trees, but random forests correct for decision trees' habit of overfitting to their training set, which can be used for classification and regression. When we carry out the classification task, new input samples will enter, and each decision tree in the forest will be allowed to judge and classify separately. Each decision tree will get its own classification result. Whichever of the classification results of the decision tree has the most classification, then the random forest will take this result as the final result. Random forest models are one of the best “off-the-shelf” prediction methods.

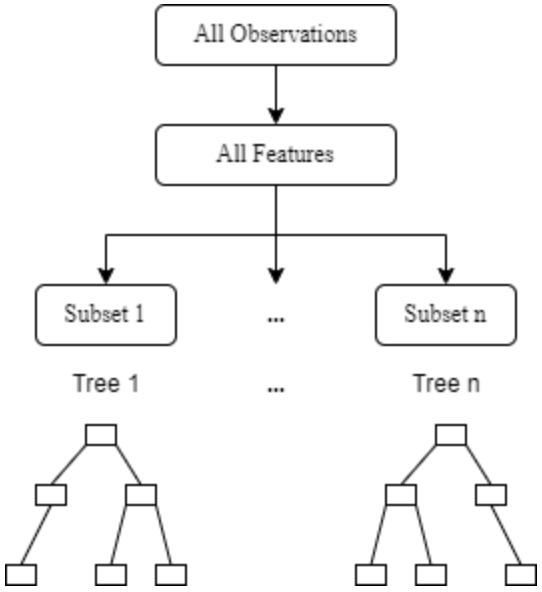


figure9

The step of doing random forest is the same as in bagging, except in addition to using many bootstrapped training datasets when deciding any split for any tree, I only use a random subset of all available predictors. Compared to other models, the random forest model has the advantage of being generally very accurate and easy to use out-of-box for classification and regression.

In this project, we use the package from the “ranger” and fit a random forest model on the train data, the ranger package is a fast implementation of random forests, particularly suited for high dimensional data.

As the fitting model shows above, we use 1000 trees, ensuring that both respect unordered factors and probability are TRUE. We put other settings to the default values. The over-average ratio is the dependent variable that we want to find. After fitting the random forest model, I got the predicted length is 21348 and has 11 forests. After that, we put the random forest to validate and test the AUC. Our results was AUC = 0.9834877, which shows the highest AUC, 98.3%, of all of the regression models. The change of AUC of the random forest after 10 times as shown in Figure 5.

5.1.5. Neural Network model

Artificial neural networks consist of many nodes interconnected with each other. Each node represents a specific output function. Each connection between two nodes represents a weighted value for the signal passing through that connection, called a weight.

In general, a neural network consists of three basic components: an input layer, a hidden layer, and an output layer.

The first of these is the input layer, which refers to the input features. In the model of this paper, to be consistent with the initial formula of several other prediction models, we select the remaining 99 features except for over_ave_ratio as the input layer, which includes the historical transaction information of bid, offer, last sold price, gender information, the attributes contained in the punk, and the overall RGBA features generated from the image pixel color information.

The hidden layer can be represented as several circles. It is called a hidden layer because in the training set, the real values of these intermediate nodes are not known to us, and their values cannot be found in the training set. For any specific circle in the hidden layer, the computation method is shown in the figure below.

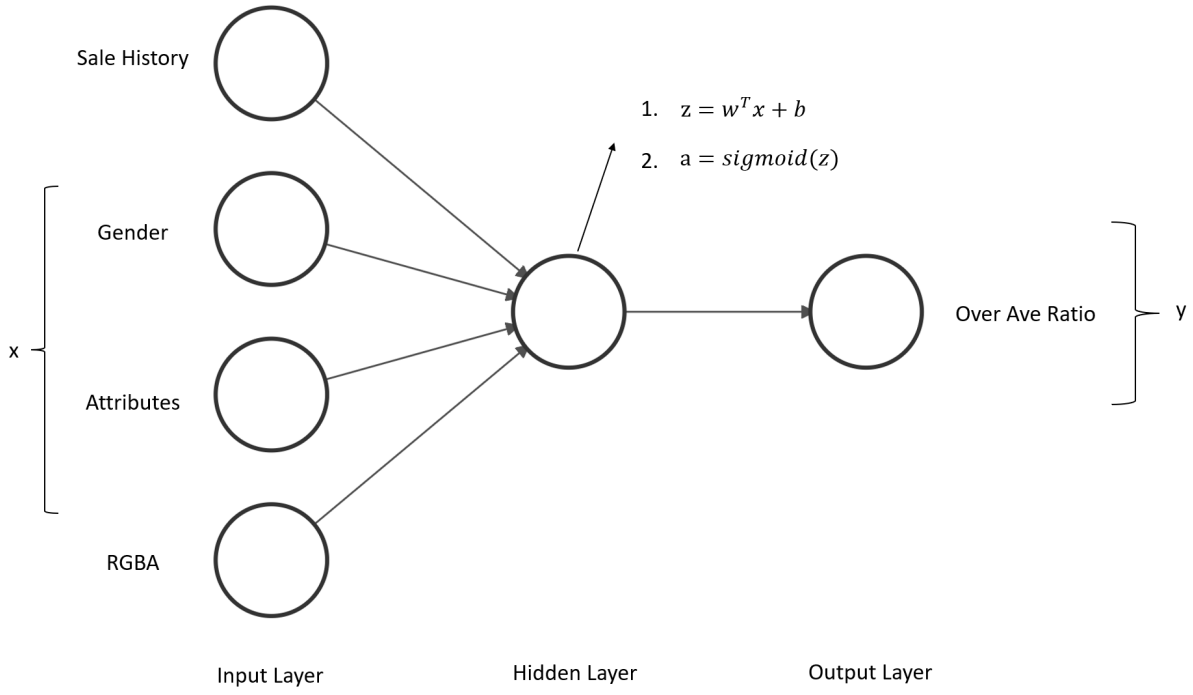


figure10

The two steps in the circle are: compute z; compute the activation function, where w and b are the parameters that the model needs to learn.

The activation function used in this neural network model is

$$sigmoid(z) = 1/(1 + e^{-z})$$

Because the value range of the Sigmoid function is restricted to between (0,1), for the judgment of binary variables, this activation function can be well associated with probability. From the steps shown in the figure, it can be seen that the computational process of the neural network is a repetition of multiple logistic regression calculations.

The last layer, called the output layer of the neural network, is responsible for outputting the predicted values. In this model, consistent with other models keeping the formula, over_ave_ratio is used as the target for the output prediction. The actual output value is the probability of over_ave_ratio=1 for that observation.

We use the package “nnet” to construct our single-hidden-layer neural network. We set the number of units in the hidden layer as 10, the maximum allowable number of weights as 10000, and the maximum number of iterations as 1000.

After fitting the model to our data, we get the model AUC as 0.9611416 and the accuracy is 0.9336582. The weights example is shown in the figure below. The red line represents positive weight, the blue line represents negative weight. The opacity of the color represents the absolute value of the weight.

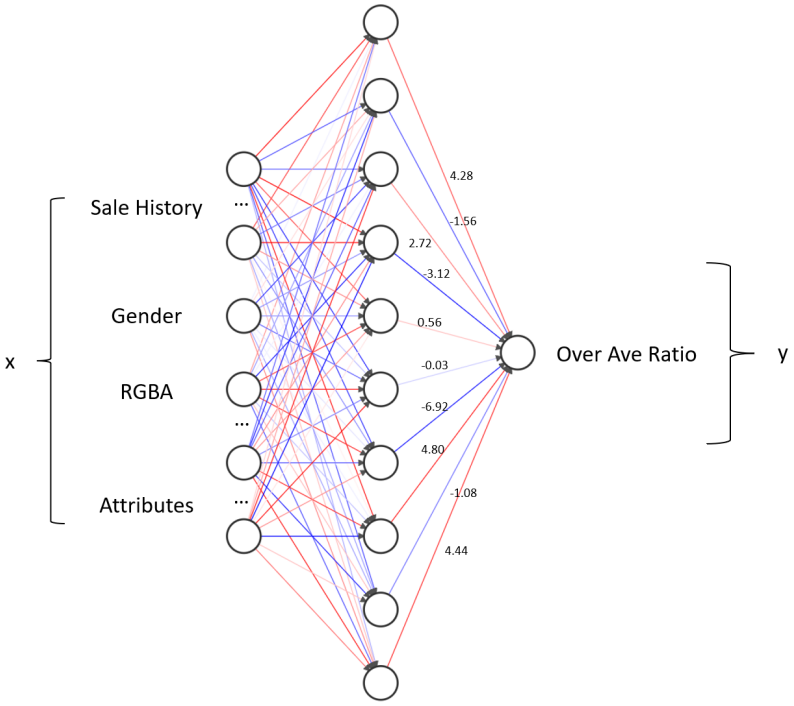


figure11

5.2. Model Evaluation and Comparison

Based on the result above, we can present the accuracy (under the threshold of 0.5) and AUC of all the models.

Model	Neural Network	Random Forest	Logistic Regression	Linear Regression	K-NN (K=24)
Accuracy	0.9336582	0.9452774	0.8952399	0.7522489	0.9124813
AUC	0.9611416	0.9834877	0.9610863	0.9627496	

From the table, we can see that the Random Forest Model provides the best result. However, when reproducing the fitting procedure, the AUC of this model can vary quite differently. So, in order to evaluate the model performance more accurately and to prevent the performance of the model occurred by chance, we bootstrap the data from the original data and rerun the models 100 times to check the performance consistency.

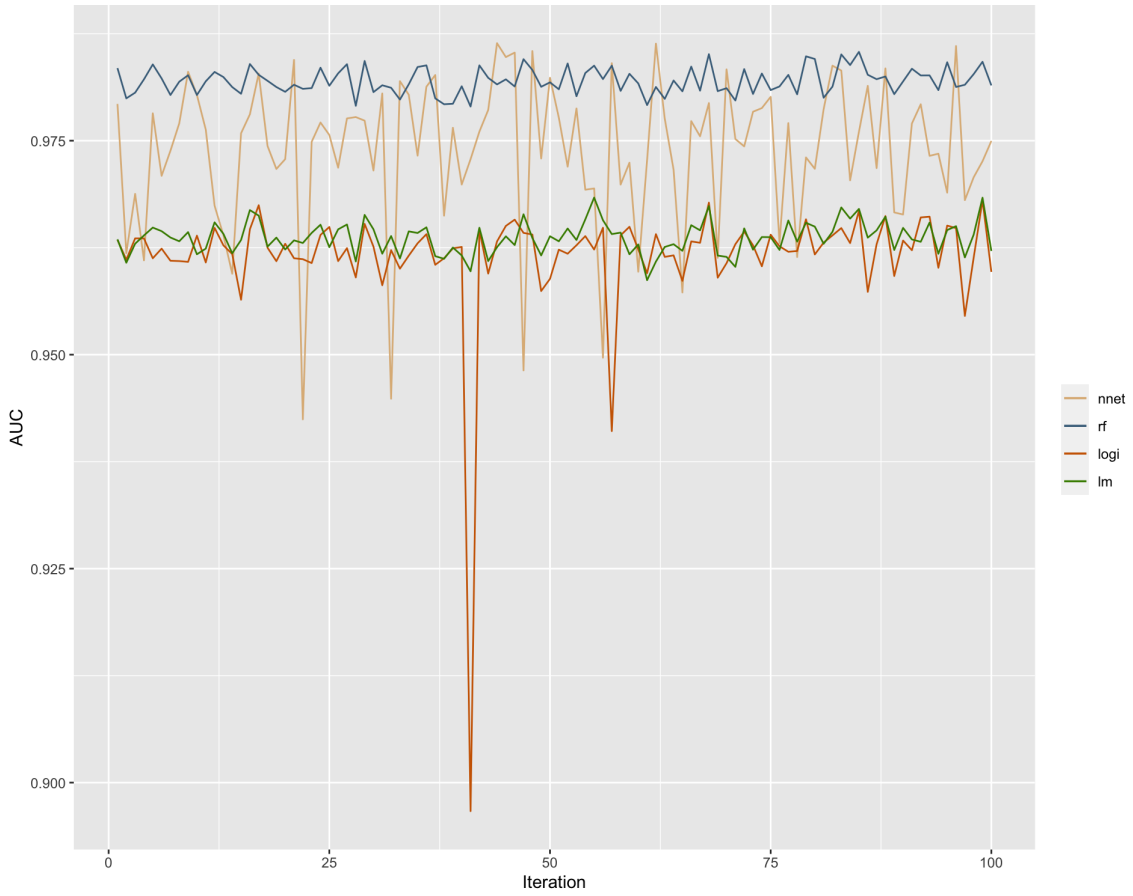


figure12

From the result figure, we can see that in over 100 times experiments, the Random Forest model showed great robustness. In general, it also has the highest AUC.

In this way, it is safe to say that the Random Forest model has the best performance based on our training data. After the model selection, we combine the training group data and validate group data to form a new training data. The random forest model based on this training data set has an AUC of 0.982519 on the testing data set.

5.3. Model Exploration

In the exploration part, our original model predict is binomial, we want to find out the real NFT price change based on our model. Thus, we transfer the result and let our model directly predict the price change of NFT. It is a more intuitive way to present the predicted results. We generate two plots, the first one is the prediction plot and the second one is the residual plot.

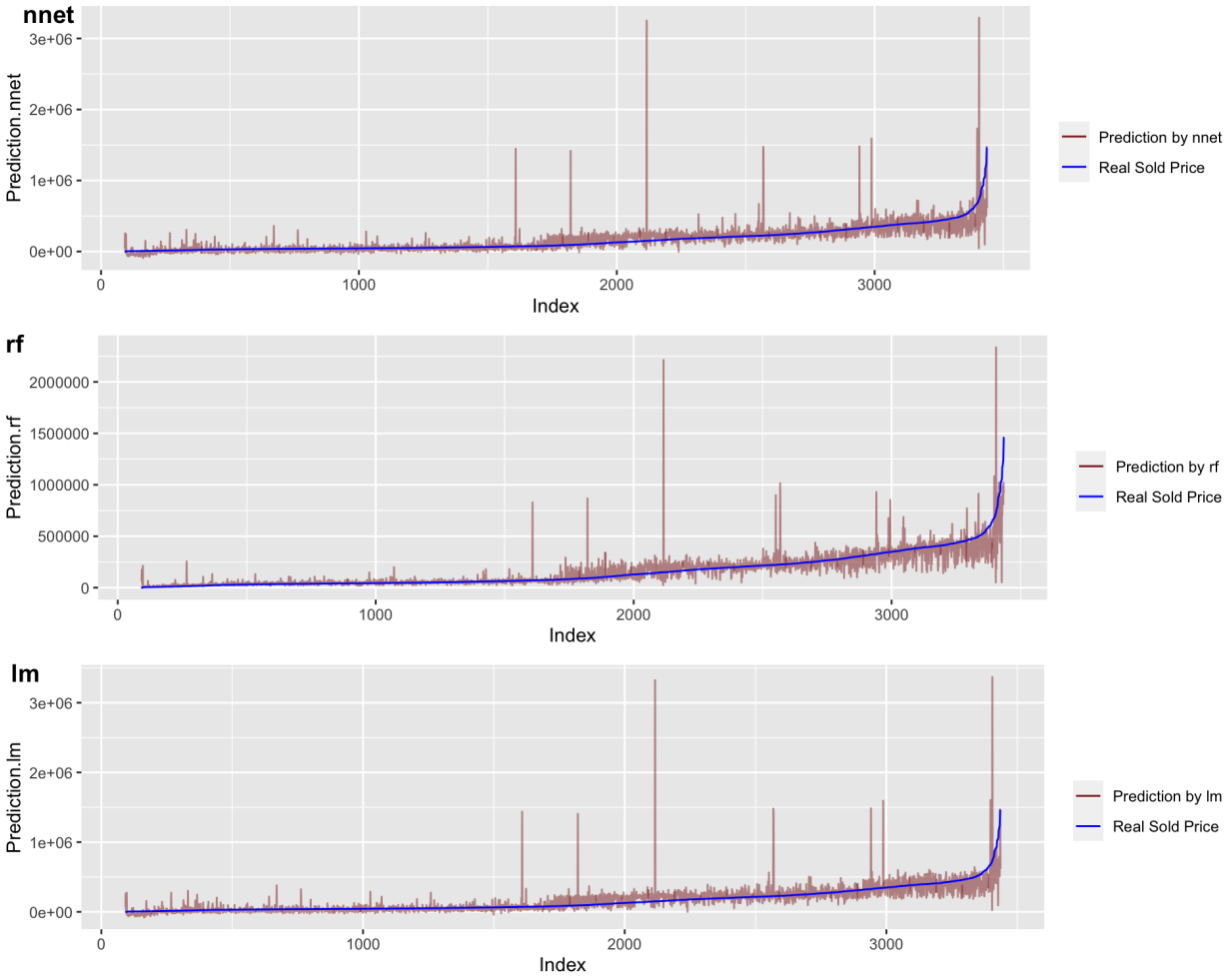


figure13 Prediction plot

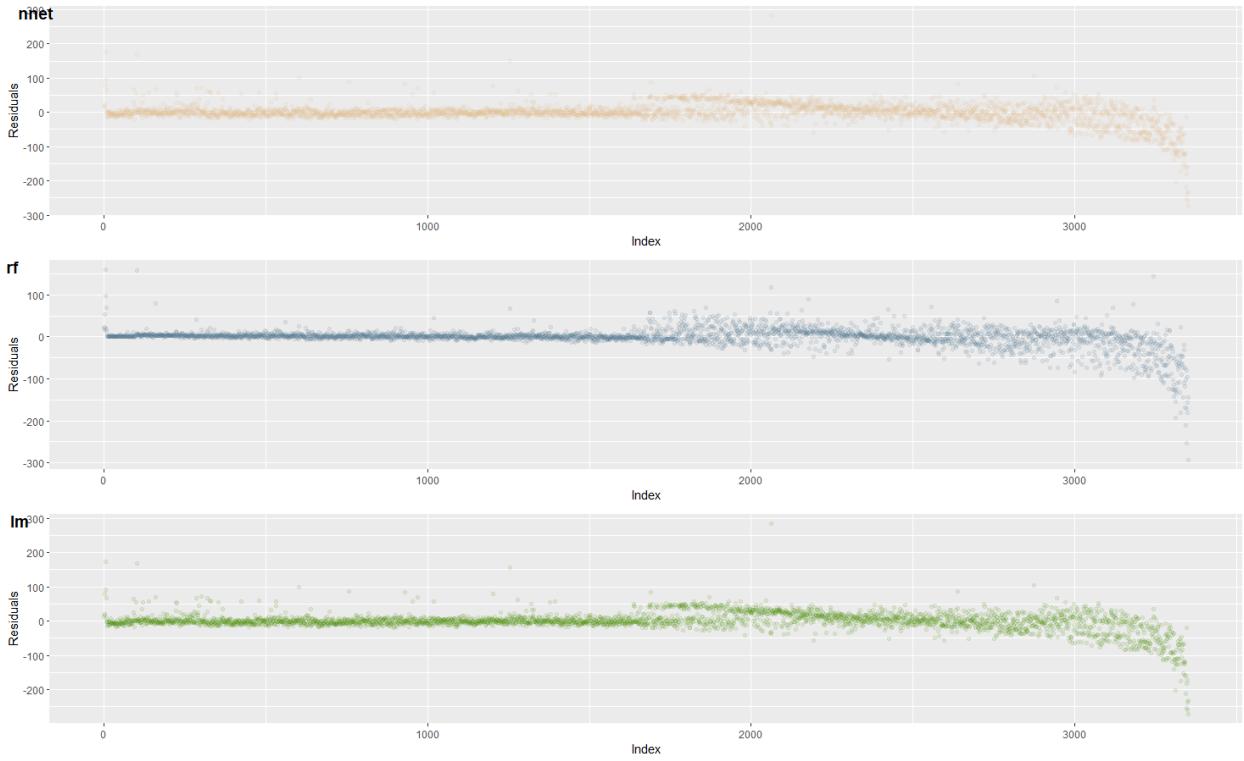


figure14 Residual plot(transformed as `ETH price`)

	Neural Network	Random Forest	Linear Regression
RSS	797.9299	592.2537	805.2255

We first compare the prediction plot of the three models, the K-NN model, the random forest model, and the linear regression model. We choose the year starting from 2021 to compare. We find out that when the index is smaller than 1000, the prediction of the random forest model is closer to the real line than others. However, after the index is over 1000, we find out that the three models all exist with some odd numbers that are higher or lower than the real sold price. It may be because of the ETH price change so we added a residual plot and transferred as ETH price. After that, we found out that the Random Forest model has the smallest residual sum of squares, which is 592.2537, and the Random Forest model also has the most concentrated residual points from figure 7. Thus, we chose the Random Forest model as the best fit model for this project.

6. Conclusion

In conclusion, comparing all the models we use, the K-Nearest Neighbors model, Linear Regression model, Logistic Regression model, Random Forest model and Neural Network model, we find out that Random Forest model has the average highest and stable AUC compared to all other models. It is higher than 0.975 which means the accurate of predict is higher than 97.5%. As the figure 5 shows, if we compare the highest AUC, we can find out that the neural network regression model may have the highest AUC, however, the neural network regression model is not stable enough. Thus, the neural network is better than logistic regression and linear regression, but worse than the Random Forest model. The logistic regression and linear regression model are showing the similar AUC, but the logistic regression exists in some AUC in very small situations. Thus, we decided to choose the Random Forest model. After that, we combined the training group data and validated the group data to be new training data. We then put the Random Forest model into the new training dataset. We tested the new Random Forest model and find out the real AUC is 0.982519. With digital currency and NFT markets gaining popularity, our group wants to use the Random Forest model we got to predict the NFT price change in the future. We only need to know the different independent variable information of NFT we want to predict, such as the last sold price, the gender difference, the attributes, the year, and the month. After that, the model can help investors decide which NFT they should invest in to ensure a decent return and the likelihood that NFT will meet their target price.

7. Bibliography

Cryptopunks. Larva Labs. (n.d.). Retrieved May 5, 2022, from https://www.larvalabs.com/cryptopunks

Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

Dobson, A. J. (1990) An Introduction to Generalized Linear Models. London: Chapman and Hall.

Raynor de Best (2022, Apr, 7), Average daily gas price of Ethereum from August 2015 to April 6, 2022. Retrieved from:https://www.statista.com/statistics/1221821/gas-price-ethereum/

Ripley, B. D. (1996) Pattern Recognition and Neural Networks. Cambridge.

The Statistics Portal. Statista. (n.d.). Retrieved May 5, 2022, from https://www.statista.com/

Venables, W. N., and Ripley, B. D. (2002). *Modern applied statistics with S*. Springer.

Wright, M. N. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw 77:1-17. doi: 10.18637/jss.v077.i01.