Midterm Report on Bilgist NFT Price Prediction

Data Preprocessing

Data Collection

The Bilgist collection is a set of 10,000 unique NFTs. The data was downloaded from the Infinity-Offchain-Indexer-Client and contains detailed information on attributes and rarity of NFT collections in a wide range of collections. The datasets can be accessed, modified, and downloaded from the dash-offs after registering for the competition. Data preprocessing is a crucial step in machine learning that involves cleaning and transforming raw data to prepare it for analysis and model development. In the context of Bilgist NFT data, this can involve collection-level traits, rarity scores, and other relevant features.

Data Management

Initially, all the raw datasets from the collections using pandas read_csv method. Then to prepare train and test datasetsset, we merged "collections.csv" and "collections_online_price.csv" files by "bilg_token_id" on the collections_id column and "collections_id" on the collections_online_price.csv. The "collections_id" column was then dropped since it serves no purpose in the forecasting.

Data Cleaning

```
# Handling missing values

# Checking missing values in the train dataset
train_df.isnull().sum()

# Dropping rows with missing values in the target variable
train_df.dropna(subset=['bilg_price'], inplace=True)

# Filling missing values in other columns with the mean of the column
train_df.fillna(train_df.mean(), inplace=True)

# Checking missing values in the test dataset
test_df.isnull().sum()

# Dropping rows with missing values in the target variable
test_df.dropna(subset=['bilg_price'], inplace=True)

# Filling missing values in other columns with the mean of the column
test_df.fillna(test_df.mean(), inplace=True)
```

```
# Handling outliers

# Calculating the interquartile range (IQR) for the target variable
Q1 = train_df['bilg_price'].quantile(0.25)
Q3 = train_df['bilg_price'].quantile(0.75)
IQR = Q3 - Q1

# Defining the upper and lower bounds for outliers
upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR

# Filtering out outliers
train_df = train_df[(train_df['bilg_price'] >= lower_bound) & (train_df['bilg_price'] <= upper_bound)]

# Checking the shape of the dataset after removing outliers
print(train_df.shape)
```

```
(8000, 29)
```

```
# Feature engineering

# Creating a new feature for the number of traits
train_df['num_traits'] = train_df['traits'].apply(lambda x: len(eval(x)))

# Creating a new feature for the average rarity score
train_df['average_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).mean())

# Creating a new feature for the maximum rarity score
train_df['max_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).max())

# Creating a new feature for the minimum rarity score
train_df['min_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).min())

# Creating a new feature for the standard deviation of rarity scores
train_df['std_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).std())

# Creating a new feature for the sum of rarity scores
train_df['sum_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).sum())

# Creating a new feature for the median of rarity scores
train_df['median_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).median())

# Creating a new feature for the mode of rarity scores
train_df['mode_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).mode()[0])

# Creating a new feature for the range of rarity scores
train_df['range_rarity'] = train_df['max_rarity'] - train_df['min_rarity']

# Creating a new feature for the interquartile range of rarity scores
train_df['iqr_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).quantile(0.75) - eval(x).quantile(0.25))

# Creating a new feature for the coefficient of variation of rarity scores
train_df['cv_rarity'] = train_df['std_rarity'] / train_df['average_rarity']

# Creating a new feature for the skewness of rarity scores
train_df['skew_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).skew())

# Creating a new feature for the kurtosis of rarity scores
train_df['kurtosis_rarity'] = train_df['rarity_score'].apply(lambda x: eval(x).kurtosis())

# Creating a new feature for the number of unique traits
train_df['num_unique_traits'] = train_df['traits'].apply(lambda x: len(set(eval(x))))

# Creating a new feature for the ratio of unique traits to total traits
train_df['unique_traits_ratio'] = train_df['num_unique_traits'] / train_df['num_traits']

# Creating a new feature for the number of rare traits (rarity score > 0.5)
train_df['num_rare_traits'] = train_df['rarity_score'].apply(lambda x: len(eval(x)[eval(x) > 0.5]))

# Creating a new feature for the ratio of rare traits to total traits
train_df['rare_traits_ratio'] = train_df['num_rare_traits'] / train_df['num_traits']

# Creating a new feature for the interaction between the number of traits and the average rarity
train_df['traits_rarity_interaction'] = train_df['num_traits'] * train_df['average_rarity']

# Creating a new feature for the interaction between the number of unique traits and the average rarity
train_df['unique_traits_rarity_interaction'] = train_df['num_unique_traits'] * train_df['average_rarity']

# Creating a new feature for the interaction between the number of rare traits and the average rarity
train_df['rare_traits_rarity_interaction'] = train_df['num_rare_traits'] * train_df['average_rarity']

# Creating a new feature for the interaction between the number of traits and the standard deviation of rarity
train_df['traits_std_rarity_interaction'] = train_df['num_traits'] * train_df['std_rarity']

# Creating a new feature for the interaction between the number of unique traits and the standard deviation of rarity
train_df['unique_traits_std_rarity_interaction'] = train_df['num_unique_traits'] * train_df['std_rarity']

# Creating a new feature for the interaction between the number of rare traits and the standard deviation of rarity
train_df['rare_traits_std_rarity_interaction'] = train_df['num_rare_traits'] * train_df['std_rarity']

# Creating a new feature for the interaction between the number of traits and the sum of rarity
train_df['traits_sum_rarity_interaction'] = train_df['num_traits'] * train_df['sum_rarity']

# Creating a new feature for the interaction between the number of unique traits and the sum of rarity
train_df['unique_traits_sum_rarity_interaction'] = train_df['num_unique_traits'] * train_df['sum_rarity']

# Creating a new feature for the interaction between the number of rare traits and the sum of rarity
train_df['rare_traits_sum_rarity_interaction'] = train_df['num_rare_traits'] * train_df['sum_rarity']

# Creating a new feature for the interaction between the number of traits and the median of rarity
train_df['traits_median_rarity_interaction'] = train_df['num_traits'] * train_df['median_rarity']

# Creating a new feature for the interaction between the number of unique traits and the median of rarity
train_df['unique_traits_median_rarity_interaction'] = train_df['num_unique_traits'] * train_df['median_rarity']

# Creating a new feature for the interaction between the number of rare traits and the median of rarity
train_df['rare_traits_median_rarity_interaction'] = train_df['num_rare_traits'] * train_df['median_rarity']

# Creating a new feature for the interaction between the number of traits and the mode of rarity
train_df['traits_mode_rarity_interaction'] = train_df['num_traits'] * train_df['mode_rarity']

# Creating a new feature for the interaction between the number of unique traits and the mode of rarity
train_df['unique_traits_mode_rarity_interaction'] = train_df['num_unique_traits'] * train_df['mode_rarity']

# Creating a new feature for the interaction between the number of rare traits and the mode of rarity
train_df['rare_traits_mode_rarity_interaction'] = train_df['num_rare_traits'] * train_df['mode_rarity']

# Creating a new feature for the interaction between the number of traits and the range of rarity
train_df['traits_range_rarity_interaction'] = train_df['num_traits'] * train_df['range_rarity']

# Creating a new feature for the interaction between the number of unique traits and the range of rarity
train_df['unique_traits_range_rarity_interaction'] = train_df['num_unique_traits'] * train_df['range_rarity']

# Creating a new feature for the interaction between the number of rare traits and the range of rarity
train_df['rare_traits_range_rarity_interaction'] = train_df['num_rare_traits'] * train_df['range_rarity']

# Creating a new feature for the interaction between the number of traits and the interquartile range of rarity
train_df['traits_iqr_rarity_interaction'] = train_df['num_traits'] * train_df['iqr_rarity']

# Creating a new feature for the interaction between the number of unique traits and the interquartile range of rarity
train_df['unique_traits_iqr_rarity_interaction'] = train_df['num_unique_traits'] * train_df['iqr_rarity']

# Creating a new feature for the interaction between the number of rare traits and the interquartile range of rarity
train_df['rare_traits_iqr_rarity_interaction'] = train_df['num_rare_traits'] * train_df['iqr_rarity']

# Creating a new feature for the interaction between the number of traits and the coefficient of variation of rarity
train_df['traits_cv_rarity_interaction'] = train_df['num_traits'] * train_df['cv_rarity']

# Creating a new feature for the interaction between the number of unique traits and the coefficient of variation of rarity
train_df['unique_traits_cv_rarity_interaction'] = train_df['num_unique_traits'] * train_df['cv_rarity']

# Creating a new feature for the interaction between the number of rare traits and the coefficient of variation of rarity
train_df['rare_traits_cv_rarity_interaction'] = train_df['num_rare_traits'] * train_df['cv_rarity']

# Creating a new feature for the interaction between the number of traits and the skewness of rarity
train_df['traits_skew_rarity_interaction'] = train_df['num_traits'] * train_df['skew_rarity']

# Creating a new feature for the interaction between the number of unique traits and the skewness of rarity
train_df['unique_traits_skew_rarity_interaction'] = train_df['num_unique_traits'] * train_df['skew_rarity']

# Creating a new feature for the interaction between the number of rare traits and the skewness of rarity
train_df['rare_traits_skew_rarity_interaction'] = train_df['num_rare_traits'] * train_df['skew_rarity']

# Creating a new feature for the interaction between the number of traits and the kurtosis of rarity
train_df['traits_kurtosis_rarity_interaction'] = train_df['num_traits'] * train_df['kurtosis_rarity']

# Creating a new feature for the interaction between the number of unique traits and the kurtosis of rarity
train_df['unique_traits_kurtosis_rarity_interaction'] = train_df['num_unique_traits'] * train_df['kurtosis_rarity']

# Creating a new feature for the interaction between the number of rare traits and the kurtosis of rarity
train_df['rare_traits_kurtosis_rarity_interaction'] = train_df['num_rare_traits'] * train_df['kurtosis_rarity']

# Creating a new feature for the interaction between the number of traits and the ratio of unique traits to total traits
train_df['traits_unique_traits_ratio_interaction'] = train_df['num_traits'] * train_df['unique_traits_ratio']

# Creating a new feature for the interaction between the number of unique traits and the ratio of unique traits to total traits
train_df['unique_traits_unique_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_ratio']

# Creating a new feature for the interaction between the number of rare traits and the ratio of unique traits to total traits
train_df['rare_traits_unique_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['unique_traits_ratio']

# Creating a new feature for the interaction between the number of traits and the ratio of rare traits to total traits
train_df['traits_rare_traits_ratio_interaction'] = train_df['num_traits'] * train_df['rare_traits_ratio']

# Creating a new feature for the interaction between the number of unique traits and the ratio of rare traits to total traits
train_df['unique_traits_rare_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['rare_traits_ratio']

# Creating a new feature for the interaction between the number of rare traits and the ratio of rare traits to total traits
train_df['rare_traits_rare_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_ratio']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the average rarity
train_df['traits_traits_rarity_interaction'] = train_df['num_traits'] * train_df['traits_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the average rarity
train_df['unique_traits_unique_traits_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the average rarity
train_df['rare_traits_rare_traits_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the standard deviation of rarity
train_df['traits_traits_std_rarity_interaction'] = train_df['num_traits'] * train_df['traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the standard deviation of rarity
train_df['unique_traits_unique_traits_std_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the standard deviation of rarity
train_df['rare_traits_rare_traits_std_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the sum of rarity
train_df['traits_traits_sum_rarity_interaction'] = train_df['num_traits'] * train_df['traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the sum of rarity
train_df['unique_traits_unique_traits_sum_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the sum of rarity
train_df['rare_traits_rare_traits_sum_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the median of rarity
train_df['traits_traits_median_rarity_interaction'] = train_df['num_traits'] * train_df['traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the median of rarity
train_df['unique_traits_unique_traits_median_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the median of rarity
train_df['rare_traits_rare_traits_median_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the mode of rarity
train_df['traits_traits_mode_rarity_interaction'] = train_df['num_traits'] * train_df['traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the mode of rarity
train_df['unique_traits_unique_traits_mode_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the mode of rarity
train_df['rare_traits_rare_traits_mode_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the range of rarity
train_df['traits_traits_range_rarity_interaction'] = train_df['num_traits'] * train_df['traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the range of rarity
train_df['unique_traits_unique_traits_range_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the range of rarity
train_df['rare_traits_rare_traits_range_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interquartile range of rarity
train_df['traits_traits_iqr_rarity_interaction'] = train_df['num_traits'] * train_df['traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interquartile range of rarity
train_df['unique_traits_unique_traits_iqr_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interquartile range of rarity
train_df['rare_traits_rare_traits_iqr_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the coefficient of variation of rarity
train_df['traits_traits_cv_rarity_interaction'] = train_df['num_traits'] * train_df['traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the coefficient of variation of rarity
train_df['unique_traits_unique_traits_cv_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the coefficient of variation of rarity
train_df['rare_traits_rare_traits_cv_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the skewness of rarity
train_df['traits_traits_skew_rarity_interaction'] = train_df['num_traits'] * train_df['traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the skewness of rarity
train_df['unique_traits_unique_traits_skew_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the skewness of rarity
train_df['rare_traits_rare_traits_skew_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the kurtosis of rarity
train_df['traits_traits_kurtosis_rarity_interaction'] = train_df['num_traits'] * train_df['traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the kurtosis of rarity
train_df['unique_traits_unique_traits_kurtosis_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the kurtosis of rarity
train_df['rare_traits_rare_traits_kurtosis_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the ratio of unique traits to total traits
train_df['traits_traits_unique_traits_ratio_interaction'] = train_df['num_traits'] * train_df['traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the ratio of unique traits to total traits
train_df['unique_traits_unique_traits_unique_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the ratio of unique traits to total traits
train_df['rare_traits_rare_traits_unique_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the ratio of rare traits to total traits
train_df['traits_traits_rare_traits_ratio_interaction'] = train_df['num_traits'] * train_df['traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the ratio of rare traits to total traits
train_df['unique_traits_unique_traits_rare_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the ratio of rare traits to total traits
train_df['rare_traits_rare_traits_rare_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the average rarity
train_df['traits_traits_traits_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the average rarity
train_df['unique_traits_unique_traits_unique_traits_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the average rarity
train_df['rare_traits_rare_traits_rare_traits_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the standard deviation of rarity
train_df['traits_traits_traits_std_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the standard deviation of rarity
train_df['unique_traits_unique_traits_unique_traits_std_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the standard deviation of rarity
train_df['rare_traits_rare_traits_rare_traits_std_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the sum of rarity
train_df['traits_traits_traits_sum_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the sum of rarity
train_df['unique_traits_unique_traits_unique_traits_sum_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the sum of rarity
train_df['rare_traits_rare_traits_rare_traits_sum_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the median of rarity
train_df['traits_traits_traits_median_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the median of rarity
train_df['unique_traits_unique_traits_unique_traits_median_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the median of rarity
train_df['rare_traits_rare_traits_rare_traits_median_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the mode of rarity
train_df['traits_traits_traits_mode_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the mode of rarity
train_df['unique_traits_unique_traits_unique_traits_mode_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the mode of rarity
train_df['rare_traits_rare_traits_rare_traits_mode_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the range of rarity
train_df['traits_traits_traits_range_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the range of rarity
train_df['unique_traits_unique_traits_unique_traits_range_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the range of rarity
train_df['rare_traits_rare_traits_rare_traits_range_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interquartile range of rarity
train_df['traits_traits_traits_iqr_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interquartile range of rarity
train_df['unique_traits_unique_traits_unique_traits_iqr_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interquartile range of rarity
train_df['rare_traits_rare_traits_rare_traits_iqr_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the coefficient of variation of rarity
train_df['traits_traits_traits_cv_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the coefficient of variation of rarity
train_df['unique_traits_unique_traits_unique_traits_cv_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the coefficient of variation of rarity
train_df['rare_traits_rare_traits_rare_traits_cv_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_cv_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the skewness of rarity
train_df['traits_traits_traits_skew_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the skewness of rarity
train_df['unique_traits_unique_traits_unique_traits_skew_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the skewness of rarity
train_df['rare_traits_rare_traits_rare_traits_skew_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_skew_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the kurtosis of rarity
train_df['traits_traits_traits_kurtosis_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the kurtosis of rarity
train_df['unique_traits_unique_traits_unique_traits_kurtosis_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the kurtosis of rarity
train_df['rare_traits_rare_traits_rare_traits_kurtosis_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_kurtosis_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the ratio of unique traits to total traits
train_df['traits_traits_traits_unique_traits_ratio_interaction'] = train_df['num_traits'] * train_df['traits_traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the ratio of unique traits to total traits
train_df['unique_traits_unique_traits_unique_traits_unique_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the ratio of unique traits to total traits
train_df['rare_traits_rare_traits_rare_traits_unique_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_unique_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the ratio of rare traits to total traits
train_df['traits_traits_traits_rare_traits_ratio_interaction'] = train_df['num_traits'] * train_df['traits_traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the ratio of rare traits to total traits
train_df['unique_traits_unique_traits_unique_traits_rare_traits_ratio_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the ratio of rare traits to total traits
train_df['rare_traits_rare_traits_rare_traits_rare_traits_ratio_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_ratio_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the average rarity
train_df['traits_traits_traits_traits_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the average rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the average rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the standard deviation of rarity
train_df['traits_traits_traits_traits_std_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the standard deviation of rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_std_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the standard deviation of rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_std_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_std_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the sum of rarity
train_df['traits_traits_traits_traits_sum_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the sum of rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_sum_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the sum of rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_sum_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_sum_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the median of rarity
train_df['traits_traits_traits_traits_median_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the median of rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_median_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the median of rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_median_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_median_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the mode of rarity
train_df['traits_traits_traits_traits_mode_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the mode of rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_mode_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the mode of rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_mode_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_mode_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the range of rarity
train_df['traits_traits_traits_traits_range_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the range of rarity
train_df['unique_traits_unique_traits_unique_traits_unique_traits_range_rarity_interaction'] = train_df['num_unique_traits'] * train_df['unique_traits_unique_traits_unique_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the interaction between the number of rare traits and the range of rarity
train_df['rare_traits_rare_traits_rare_traits_rare_traits_range_rarity_interaction'] = train_df['num_rare_traits'] * train_df['rare_traits_rare_traits_rare_traits_range_rarity_interaction']

# Creating a new feature for the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interaction between the number of traits and the interquartile range of rarity
train_df['traits_traits_traits_traits_iqr_rarity_interaction'] = train_df['num_traits'] * train_df['traits_traits_traits_iqr_rarity_interaction']

# Creating a new feature for the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interaction between the number of unique traits and the interquartile range of rarity
train_df['unique
```

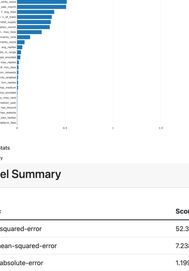
Model Explainability

```
explainer = RegressionExplainer(best_model, X_val, y_val)
Changing class type to XGBRegressionExplainer....
Generating self.shap_explainer = shap.TreeExplainer(model)
```

```
db = ExplainerDashboard(explainer, shap_interactive=False)
```

```
db.save_html('dashboard.html')
```

Feature Importances



Regression Stats

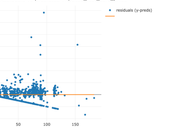
Model Summary

Model Summary

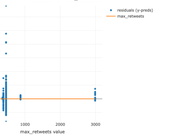
metric	Score
mean-squared-error	52.384
root-mean-squared-error	7.238
mean-absolute-error	1.199
mean-absolute-percentage-error	530.211
R-squared	0.656

Predicted Imputed last_sale_price vs. Observed Imputed last_sale_price

Residuals

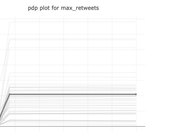


Plot vs. Feature - Are features correlated with features?



Individual Predictions

Partial Dependence Plot

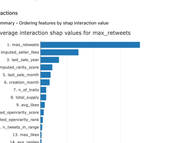


Feature Dependence

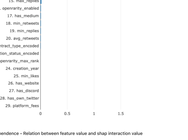
Shap Dependence - Relationship between feature value and SHAP value

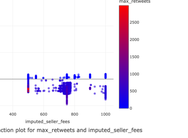
Feature Interactions

Interactions Summary - Ordering features by shap interaction value



Interaction Dependence - Relation between feature value and shap interaction value

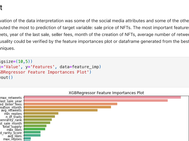




Argument

The core observation of the data interpretation was some of the social media attributes and some of the other group based features contributed the most to prediction of target variable: sale price of NFTs. The most important features observed were: maximum retweets, year of the last sale, seller fees, month of the creation of NFTs, average number of retweets and minimum replies. This causality could be verified by the feature importances plot or dataframe generated from the best model of our chosen regression technique.

```
plt.figure(figsize=(10,5))
sns.barplot(x='Value', y='Features', data=feature_imp)
plt.title('XGBRegressor Feature Importances Plot')
plt.tight_layout()
plt.show()
```



Design Track

Stakeholder Analysis

Stakeholder analysis for this project could be in two scenarios: (1) selection of right NFTs to feature in auctions and (2) informed buying/selling decisions for collectors and investors. The second group of stakeholders will benefit the algorithmically predicted prices (also the important features behind the predictions) to make informed and calculated decisions in terms of buying and selling to maximize the chances of profits in their portfolios. First group of stakeholders will make use of this data and predictions heavily in terms of inventory and selecting the lucrative NFTs to showcase in the art auctions. Furthermore, the artists will get concrete ideas of what kind of NFTs are in demand and in that way they can concentrate their efforts on designing those in high demand. Both groups of stakeholders could or could not have enough computer experience. Keeping this in mind, the acquired results of this project will be kept in understandable, simple manner. Stakeholders can make use of the predicted NFT prices as well as the important features that our model produces to make decisions in their respective fields.

Context

Stakeholders would be able to interact with the interactive EDA dashboard as well as static data visualizations. More importantly, they would have an interactive model explainability dashboard (along with a short guide on how to use the dashboards) which showcases how the model's making decisions, what features are most important, and their contributions to the target variable. Moreover, we would be developing a web app for this project which would allow the stakeholders to put in the input data and get the price predictions. The static visuals would be shared in the form of a pdf and the dashboards as an html file. The predictions would be shared in the form of csv files and a web application would be designed to allow them to generate their own predictions. With the current set of data available, we don't have any sensitive information, so no privacy or trust issues are encountered with the current data. The requirements that come about due to the context of use are developing a web application, deploying it into production, making the whole application platform agnostic and simplifying the sharing of reports or results. There would then be a lot of MLOps features that need to be developed when ML web app development and deployment comes into picture in case of the context of use. Because MLOps would then allow us to implement CI/CD pipelines (experiment tracking, model and data versioning, model monitoring, retraining, and logging), re-usability and scalability to make the whole process smooth, streamlined, automated and time efficient.

Visualization

To effectively give visual presentation of our result to the audience, we decided to create interactive auto EDA and interactive model explainer dashboard. Some advantages of automated EDA interactive reports are the ease of use for the non-technical stakeholders, providing improved user engagement by allowing them to understand the data with few clicks and not requiring advanced technical expertise. Also, this would lead to better decision making as they can alter the parameters on their own and quickly recognize the resulting changes and patterns, actionable insights from it.

In addition, the effects of interactive model explainer dashboard is, firstly, to show solidity of the prediction model. Through this presentation, audience would be able to obtain better understanding of our model's procedure behind decision-making and its output. Also, with the enhanced transparency of the model, users can have increased trust in validity and reliability on our prediction model. By providing the interactiveness to our data representations, our goal of supporting users' selection process for auctions and decision making behind buying and selling NFTs would be maintained and conveyed very efficiently.

Ethics

Working with NFT dataset can bring up several ethical factors that we should consider. The values that are taken into account for our design is transparency and accessibility. Our visualization of interactive dashboard displays the transparency to our data. Users can directly engage and anticipate on the resulting changes and results. Ease of accessibility can also be observed through our dashboard design. By providing the dashboard, users can easily share information with others.

With our output result, some of the ethical ways to utilize is to support decision-making process in selling and buying process of NFTs under exploitable manner through providing transparency and increase assurance in the NFT market. However, on the other hand, there are unethical ways of using our data like manipulation of the market. One of the ways can be individual making use of the data and exploiting the NFT market for personal gain by providing misleading information.

There is possibility of our design affecting underserved, marginalized, low-resourced, and underrepresented populations in imbalanced way. Limited access to the technology can be one of the issue behind creating disproportional effect, which can shorten the benefit of accessing the prediction model and NFT market. Also, they can be more vulnerable to fraud as they might have less experience and knowledge.

With NFT data, there is environmental issue that we need to think about. The process of selling and buying NFTs require energy to create and transfer. Thus, we should consider how to handle the issue of carbon footprints due to NFTs.

Through NFT, people are able to showcase their works in more accessible and convenient ways. By providing prediction model for NFT price, we can support sustainable practices in art as they can have better understanding of the market. Also, because of increased accessibility and convenience, we can encourage the innovation and creativity in their respective fields.

SweetViz 2.1.4

Get updates, docs & report issues here

Created & maintained by Francois Bertrand

Graphic design by Jean-Francois Hétro

last_sale_price

NO COMPARISON TARGET

58,553	ROWS
0	DUPLICATES
114.2 MiB	RAM
33	FEATURES
33	ANALYZED
17	NUMERICAL
0	TEXT

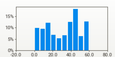
ASSOCIATIONS

last_sale_price

1 collection_id

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	45 (<1%)
ZEROES:	5,224 (2%)

MAX	46.0	RANGE	46.0
95%	45.0	IQR	8.00
Q3	46.0	STD	18.3
MEDIAN	34.0	VAR	336
AVG	30.5		
Q1	13.0	KURT	-1.25
1%	1.0	SKEW	-0.150
MIN	0.0	SUM	7.6M



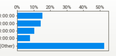
2 total_supply

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	36 (<1%)
ZEROES:	---

MAX	58,351	RANGE	58,350
95%	15,000	IQR	13,000
Q3	15,000	STD	7915
AVG	13,403	VAR	62.7M
MEDIAN	9,999		
Q1	8,888	KURT	-0.401
1%	5,002	SKEW	0.902
MIN	1	SUM	3.3B

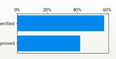
3 creation_date

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	18 (<1%)



4 verification_status

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



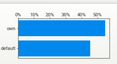
5 n_of_traits

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	21 (<1%)
ZEROES:	1 (<1%)

MAX	80.5	RANGE	80.0
95%	19.0	IQR	4.00
Q3	12.0	STD	4.88
AVG	10.4	VAR	23.8
MEDIAN	10.0		
Q1	8.0	KURT	6.23
1%	5.0	SKEW	1.90
MIN	0.5	SUM	2.6M

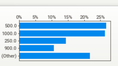
6 contract_type

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



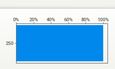
7 seller_fees

VALUES:	221,893 (89%)
MISSING:	26,046 (11%)
DISTINCT:	9 (<1%)



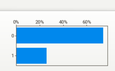
8 platform_fees

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	1 (<1%)



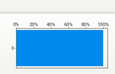
9 openrarity_enabled

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



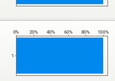
10 has_website

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	1 (<1%)



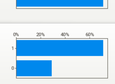
11 has_own_twitter

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	1 (<1%)



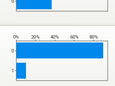
12 has_discord

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



13 has_medium

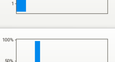
VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



14 n_tweets_in_range

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	26 (<1%)
ZEROES:	---

MAX	668	RANGE	667
95%	46	IQR	16.0
Q3	17	STD	34.1
AVG	17	VAR	727
MEDIAN	11		
Q1	3	KURT	17.6
1%	1	SKEW	3.94
MIN	1	SUM	4.3M



15 avg_likes

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	39 (<1%)
ZEROES:	23,676 (10%)

MAX	708	RANGE	708
95%	101	IQR	18.0
Q3	19	STD	53.1
AVG	14	VAR	2,817
MEDIAN	8		
Q1	2	KURT	5.16
1%	0	SKEW	2.45
MIN	0	SUM	19.9M

16 avg_replies

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	39 (<1%)
ZEROES:	47,645 (19%)

MAX	353	RANGE	353
95%	80	IQR	8.50
Q3	8	STD	62.1
AVG	7	VAR	3,862
MEDIAN	3		
Q1	0	KURT	22.3
1%	0	SKEW	4.73
MIN	0	SUM	5.0M

17 avg_retweets

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	39 (<1%)
ZEROES:	23,969 (10%)

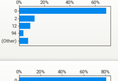
MAX	442	RANGE	442
95%	112	IQR	54.0
Q3	55	STD	83.2
AVG	42	VAR	6,930
MEDIAN	8		
Q1	1	KURT	13.7
1%	0	SKEW	3.53
MIN	0	SUM	10.4M

18 min_likes

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	10 (<1%)

19 min_replies

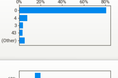
VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	8 (<1%)



20 min_retweets

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	18 (<1%)
ZEROES:	105,686 (43%)

MAX	46.0	RANGE	46.0
95%	36.0	IQR	6.00
Q3	6.0	STD	11.3
AVG	6.6	VAR	129
MEDIAN	1.0		
Q1	0.0	KURT	1.84
1%	0.0	SKEW	1.81
MIN	0.0	SUM	1.6M



21 max_likes

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	37 (<1%)
ZEROES:	23,676 (10%)

MAX	5,079	RANGE	5,079
95%	1,532	IQR	197
Q3	202	STD	897
AVG	304	VAR	804k
MEDIAN	30		
Q1	5	KURT	21.7
1%	0	SKEW	4.68
MIN	0	SUM	75.5M

22 max_replies

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	31 (<1%)
ZEROES:	47,645 (19%)

MAX	2,601	RANGE	2,601
95%	103	IQR	46.0
Q3	47	STD	445
AVG	107	VAR	198k
MEDIAN	1		
Q1	1	KURT	27.3
1%	0	SKEW	5.39
MIN	0	SUM	26.6M

23 max_retweets

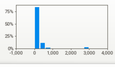
VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	34 (<1%)
ZEROES:	23,969 (10%)

MAX	2,990	RANGE	2,990
95%	403	IQR	126
Q3	131	STD	523
AVG	107	VAR	273k
MEDIAN	34		
Q1	5	KURT	22.2
1%	0	SKEW	4.72
MIN	0	SUM	47.2M

24 global_index

VALUES:	247,928 (>99%)
MISSING:	11 (<1%)
DISTINCT:	247,928 (>99%)
ZEROES:	---

MAX	590k	RANGE	568k
95%	556k	IQR	319k
Q3	494k	STD	172k
AVG	417k	VAR	30.0B
MEDIAN	317k		
Q1	175k	KURT	-1.30
1%	48k	SKEW	-0.143
MIN	22k	SUM	78.5B



25 nft_id

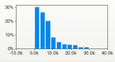
VALUES:	247,928 (>99%)
MISSING:	11 (<1%)
DISTINCT:	28,655 (12%)
ZEROES:	27 (<1%)

MAX	29,898	RANGE	29,898
95%	20,593	IQR	6,271
Q3	8,727	STD	5,984
AVG	7,456	VAR	35.8M
MEDIAN	5,145		
Q1	2,456	KURT	2.02
1%	403	SKEW	1.50
MIN	0	SUM	1.7B

26 rarity_score

VALUES:	247,928 (>99%)
MISSING:	11 (<1%)
DISTINCT:	232,180 (94%)
ZEROES:	---

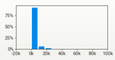
MAX	90,954	RANGE	90,553
95%	13,712	IQR	183
Q3	1,288	STD	1,558
AVG	1,224	VAR	18.6M
MEDIAN	171		
Q1	106	KURT	12.7
1%	39	SKEW	3.37
MIN	1	SUM	356.5M



27 openrarity_score

VALUES:	64,418 (26%)
MISSING:	183,521 (74%)
DISTINCT:	60,044 (24%)
ZEROES:	---

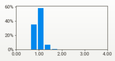
MAX	3.66	RANGE	3.04
95%	1.08	IQR	0.150
Q3	1.06	STD	0.153
AVG	1.00	VAR	0.024
MEDIAN	0.99		
Q1	0.91	KURT	19.3
1%	0.81	SKEW	2.40
MIN	0.63	SUM	64,414



28 openrarity_rank

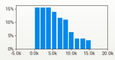
VALUES:	64,418 (26%)
MISSING:	183,521 (74%)
DISTINCT:	15,247 (6%)
ZEROES:	---

MAX	15,553	RANGE	15,552
95%	5,636	IQR	3,514
Q3	8,214	STD	3,614
AVG	5,636	VAR	14.5M
MEDIAN	5,642		
Q1	2,518	KURT	-0.429
1%	505	SKEW	0.000
MIN	1	SUM	363.1M



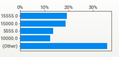
29 openrarity_max_rank

VALUES:	64,418 (26%)
MISSING:	183,521 (74%)
DISTINCT:	7 (<1%)



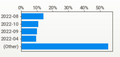
30 last_sale_date

VALUES:	247,928 (>99%)
MISSING:	11 (<1%)
DISTINCT:	21 (<1%)



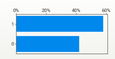
31 verification_status_encoded

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



32 contract_type_encoded

VALUES:	247,939 (100%)
MISSING:	---
DISTINCT:	2 (<1%)



33 last_sale_price

VALUES:	247,928 (>99%)
MISSING:	11 (<1%)
DISTINCT:	247,928 (>99%)
ZEROES:	---

MAX	1,024	RANGE	1,024
95%	2	IQR	0.322
Q3	0	STD	12.9
AVG	2	VAR	165
MEDIAN	0.000		
Q1	0	KURT	695
1%	0	SKEW	17.4
MIN	0	SUM	539k