Team Name

Binary Beast

Members

Bhupesh Dhurandher

Lisha Bohra

Indraja B L S

Neha Zulfiqar Khundmiri

Mentor

Dr. Chandrika M

Describe your Idea/Solution

Blockchain network users may learn more about the supply and demand of non-fungible tokens (NFTs) by using analytics tools. Currently, the NFT market is flourishing.

Unique digital objects with blockchain-managed ownership are known as NFTs. NFTs include things like digital artwork, collectibles, virtual reality gear, cryptocurrency domain names, and ownership records for tangible objects.

NFTGO data show that since May 2021, at least one NFT project has been launched on chain every day. Investors may easily fall victim to the "liquidity trap" since NFT projects' quality varies, just like DeFi's does. In truth, you may rely on statistics in the realm of blockchain to avoid "hearsay" and choose which NFT to buy, whether you're a newbie or an investment.

To learn more about NFT, analytics dashboards and reports may be made. Traders, builders, and collectors might get a competitive edge in a brand-new market by applying analytics tools to spot trends and anomalies.

Currently, NFTs based on decentralized communities are being created. Online communities have developed strategies to offer social value due to their rapid growth. A portion of the funds generated is used to advance society.

Novelty/Uniqueness

This dashboard gives investors a beautifully condensed overview of the NFT market. The most recent projects that Smart Minters are minting, the most recent volume of NFT mints, and a leaderboard of the most lucrative wallets are all displayed to investors in this tab.

To pinpoint the elements influencing an NFT's market value, this is real alpha. Investors may view the whole market and the patterns emerging within it from a single source rather than slogging through Discord groups and Twitter. Investors have access to the newest mints, and time is crucial when trading NFTs.

A summary of the NFT collections and price changes for Ethereum NFT are given in the Market Overview NFT data analytics dashboard.

This enables investors to keep tabs on NFT industry developments and determine the mood in the market. Investors may locate NFT collections that are becoming more popular by filtering the price change and minting or buying them. By using the analytics to trade with momentum, this dashboard makes trend trading possible for investors.

From a single dashboard, investors may monitor price movements, volume, and the top NFT flippers. Finding lucrative projects early on has grown more difficult due to the growth of NFTs, which is why investors need an NFT analytics platform to make sure they stay ahead of the competition.

Business Impact

Users may get useful information and insights about the market as well as particular NFT projects through the NFT Sales Analytics Dashboard.

Instead of responding on their feelings and impressions, this enables them to make better educated judgments and maybe enhance their entire trading approach.

Informational graphic defining NFT analytics tools. You may use it to access all of the open data on the blockchain and acquire insights into what is popular, who is trading what, and where profitable trade opportunities are.

Data's enormous value has been increasingly evident since the beginning of the digital era. Data is information, and you can gain from having more accurate information than others.

Technical Architecture

The uniqueness of the asset is guaranteed by the safe recording of NFTs on a block chain, which is the same technology used to create crypto currencies. It may also be challenging to modify or counterfeit NFTs due to the technology.

Cognos Analytics dashboard is used to generate the NFT sales analytics Dashboard.

Various ML techniques are used to preprocess and show the data in the csv file.

The data is obtained from the nfts.dataset and subsequently processed, with various tools being utilized to carry out particular tasks.

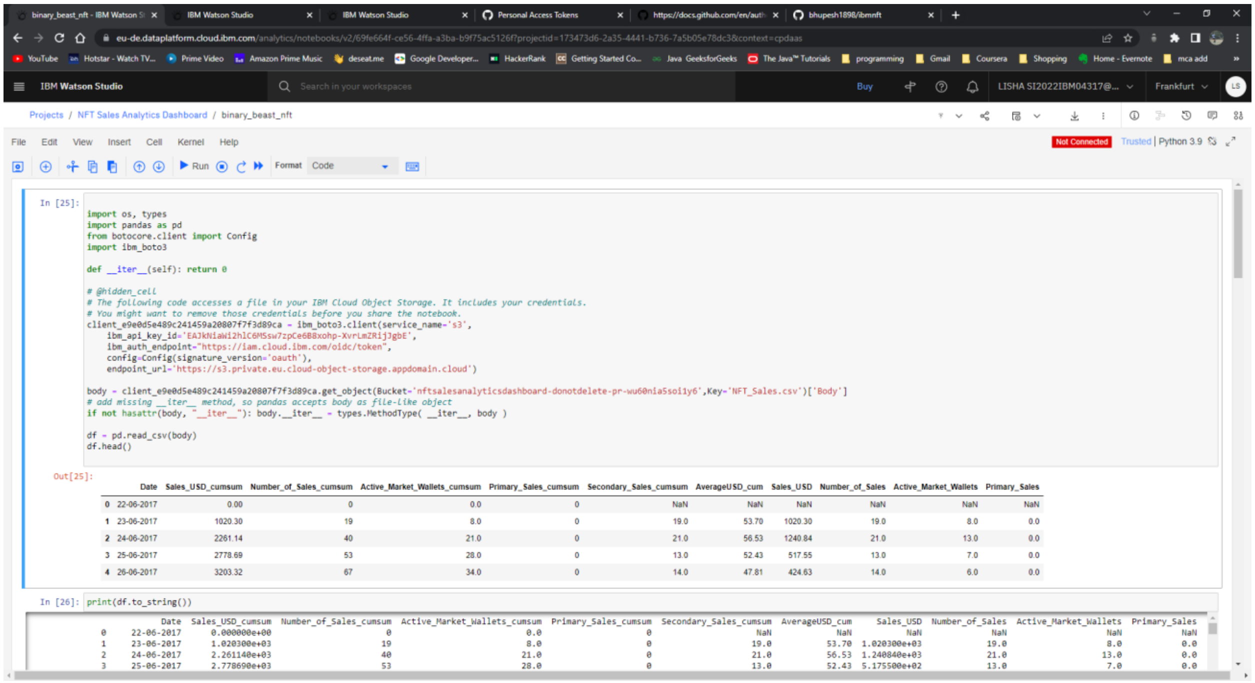
To produce an estimated statistical dataset, the graph is drawn using the dataset and the matplotlib tool.

Scope of the work

The platform's goal is to become the go-to information source for all things non-fungible, and it is developing an ecosystem of artists and curators that can add user-generated material to discover NFT data analytics.

The goal of this solution is to provide a dashboard that can view NFT Data and extract learnings that can benefit both buyers and sellers.

Applying statistical analysis and technology to data in order to identify trends and resolve issues is the main goal of data analytics. Enterprises are increasingly relying on data analytics to analyze and shape business processes, enhance decision-making, and boost financial performance.

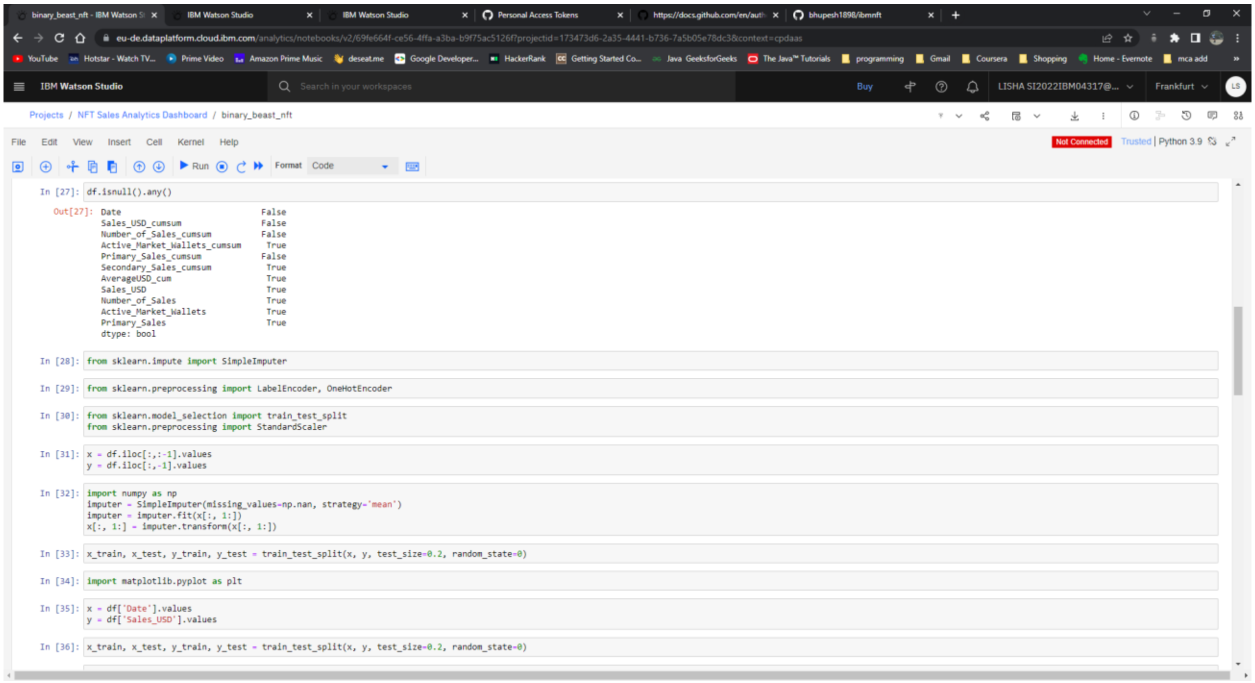


```
In [25]:
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_e9e0d5e489c241459a20807f7f3d89ca = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='EAJkNiaWi2hlC6MSsw7zpCe6B8xohp-XvrLmZRijJgbE',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

body = client_e9e0d5e489c241459a20807f7f3d89ca.get_object(Bucket='nftsalesanalyticsdashboard-donotdelete-pr-wu60nia5soi1y6',Key='NFT_Sales.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head()
```

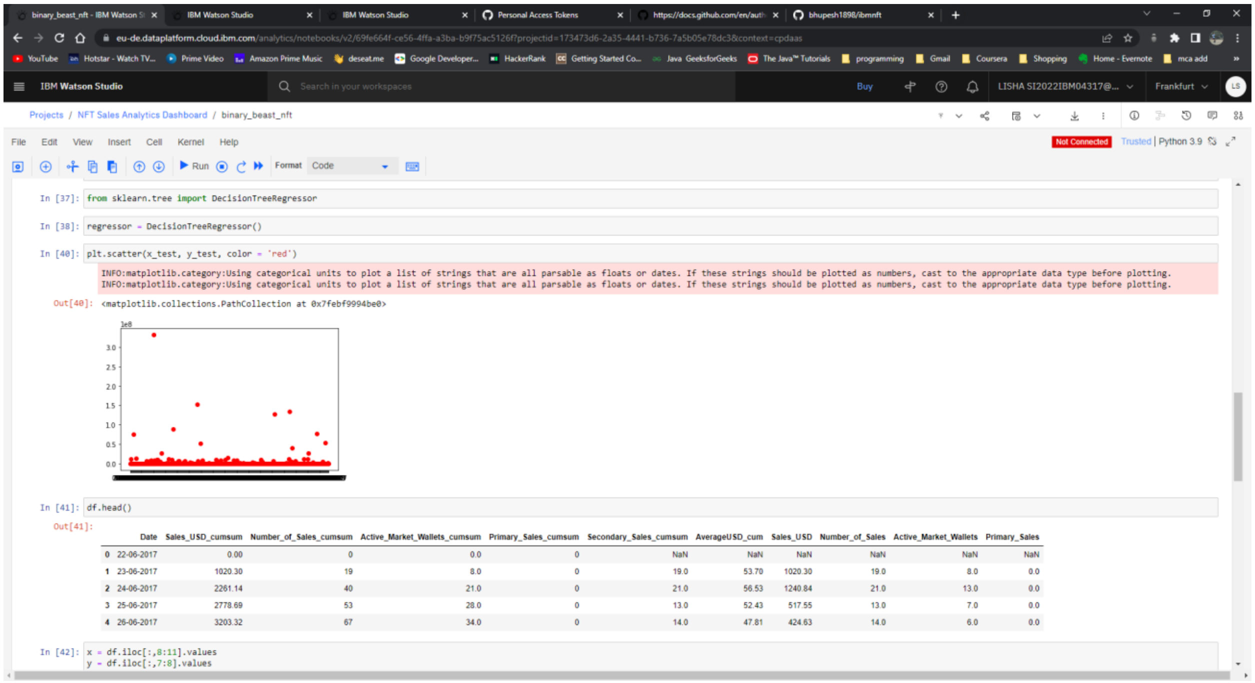
Out[25]:

	Date	Sales_USD_cumsum	Number_of_Sales_cumsum	Active_Market_Wallets_cumsum	Primary_Sales_cumsum	Secondary_Sales_cumsum	AverageUSD_cum	Sales_USD	Number_of_Sales	Active_Market_Wallets	Primary_Sales
0	22-06-2017	0.00	0	0.0	0	NaN	NaN	NaN	NaN	NaN	NaN
1	23-06-2017	1020.30	19	8.0	0	19.0	53.70	1020.30	19.0	8.0	0.0
2	24-06-2017	2261.14	40	21.0	0	21.0	56.53	1240.84	21.0	13.0	0.0
3	25-06-2017	2778.69	53	28.0	0	13.0	52.43	517.55	13.0	7.0	0.0
4	26-06-2017	3203.32	67	34.0	0	14.0	47.81	424.63	14.0	6.0	0.0

```
In [26]: print(df.to_string())

         Date  Sales_USD_cumsum  Number_of_Sales_cumsum  Active_Market_Wallets_cumsum  Primary_Sales_cumsum  Secondary_Sales_cumsum  AverageUSD_cum     Sales_USD  Number_of_Sales  Active_Market_Wallets  Primary_Sales
0  22-06-2017      0.000000e+00                       0                           0.0                     0                     NaN             NaN           NaN              NaN                    NaN            NaN
1  23-06-2017      1.020300e+03                      19                           8.0                     0                    19.0           53.70  1.020300e+03             19.0                    8.0            0.0
2  24-06-2017      2.261140e+03                      40                          21.0                     0                    21.0           56.53  1.240840e+03             21.0                   13.0            0.0
3  25-06-2017      2.778690e+03                      53                          28.0                     0                    13.0           52.43  5.175500e+02             13.0                    7.0            0.0
```


binary_beast_nft - IBM Watson St
IBM Watson Studio
IBM Watson Studio
Personal Access Tokens
https://docs.github.com/en/auth
bhupesh1898/ibmnft
eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/69fe664f-ce56-4ffa-a3ba-b9f75ac5126f?projectid=173473d6-2a35-4441-b736-7a5b05e78dc3&context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
LISHA SI2022IBM04317@...
Frankfurt
Projects / NFT Sales Analytics Dashboard / binary_beast_nft
File
Edit
View
Insert
Cell
Kernel
Help
Not Connected
Trusted
Python 3.9
Run
Format
Code

```
In [27]: df.isnull().any()

Out[27]: Date                            False
         Sales_USD_cumsum                False
         Number_of_Sales_cumsum          False
         Active_Market_Wallets_cumsum     True
         Primary_Sales_cumsum            False
         Secondary_Sales_cumsum           True
         AverageUSD_cum                   True
         Sales_USD                        True
         Number_of_Sales                  True
         Active_Market_Wallets            True
         Primary_Sales                    True
         dtype: bool

In [28]: from sklearn.impute import SimpleImputer

In [29]: from sklearn.preprocessing import LabelEncoder, OneHotEncoder

In [30]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import StandardScaler

In [31]: x = df.iloc[:,:-1].values
         y = df.iloc[:,-1].values

In [32]: import numpy as np
         imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
         imputer = imputer.fit(x[:, 1:])
         x[:, 1:] = imputer.transform(x[:, 1:])

In [33]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

In [34]: import matplotlib.pyplot as plt

In [35]: x = df['Date'].values
         y = df['Sales_USD'].values

In [36]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

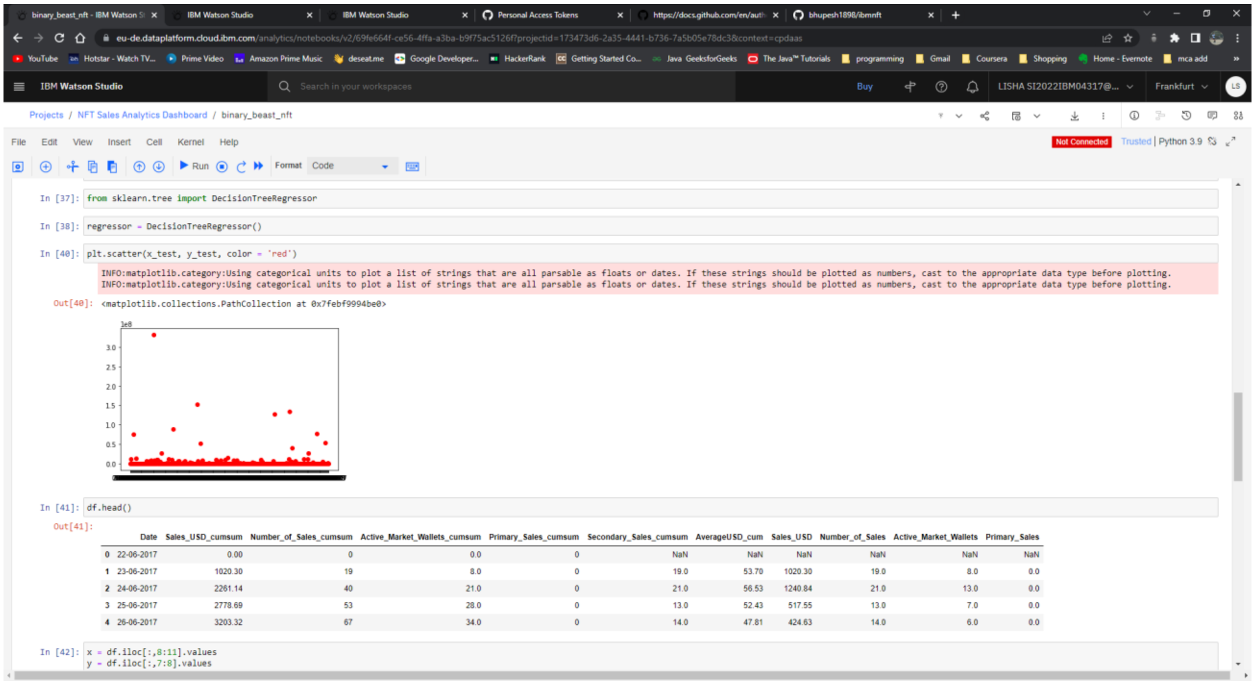


```
In [37]: from sklearn.tree import DecisionTreeRegressor

In [38]: regressor = DecisionTreeRegressor()

In [40]: plt.scatter(x_test, y_test, color = 'red')

         INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
         INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.

Out[40]: <matplotlib.collections.PathCollection at 0x7febf9994be0>
```

```
In [41]: df.head()

Out[41]:
```

	Date	Sales_USD_cumsum	Number_of_Sales_cumsum	Active_Market_Wallets_cumsum	Primary_Sales_cumsum	Secondary_Sales_cumsum	AverageUSD_cum	Sales_USD	Number_of_Sales	Active_Market_Wallets	Primary_Sales
0	22-06-2017	0.00	0	0.0	0	NaN	NaN	NaN	NaN	NaN	NaN
1	23-06-2017	1020.30	19	8.0	0	19.0	53.70	1020.30	19.0	8.0	0.0
2	24-06-2017	2261.14	40	21.0	0	21.0	56.53	1240.84	21.0	13.0	0.0
3	25-06-2017	2778.69	53	28.0	0	13.0	52.43	517.55	13.0	7.0	0.0
4	26-06-2017	3203.32	67	34.0	0	14.0	47.81	424.63	14.0	6.0	0.0

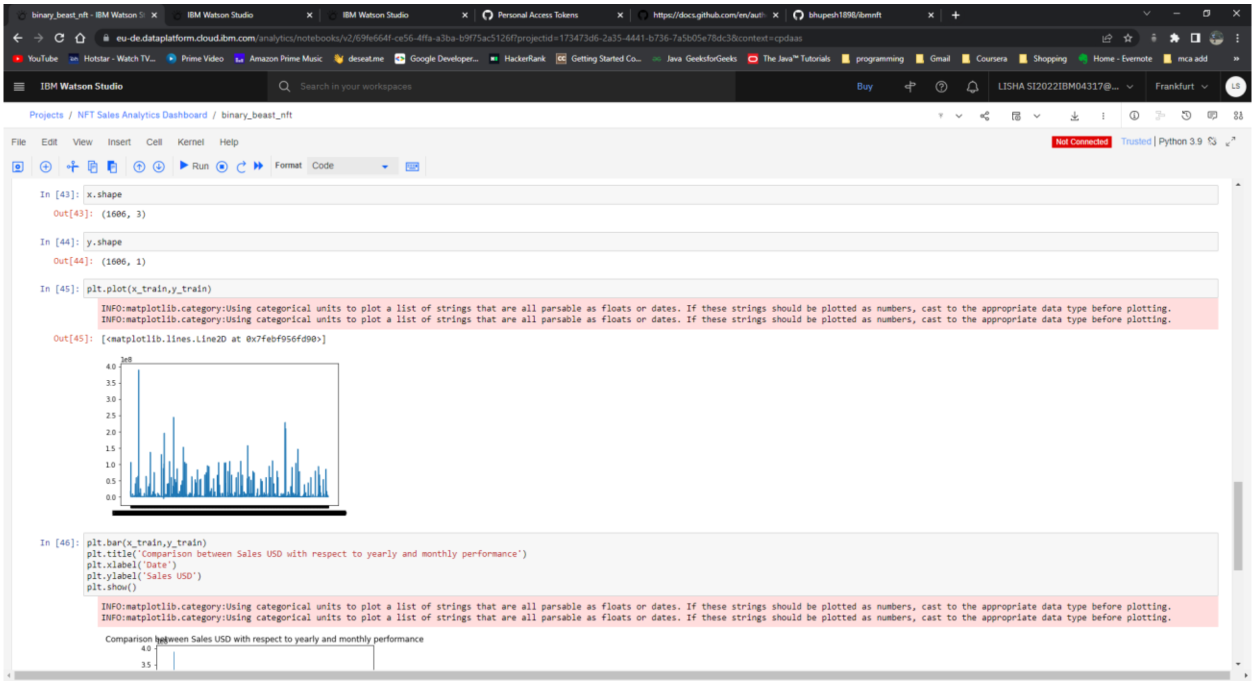
```
In [42]: x = df.iloc[:,8:11].values
         y = df.iloc[:,7:8].values
```



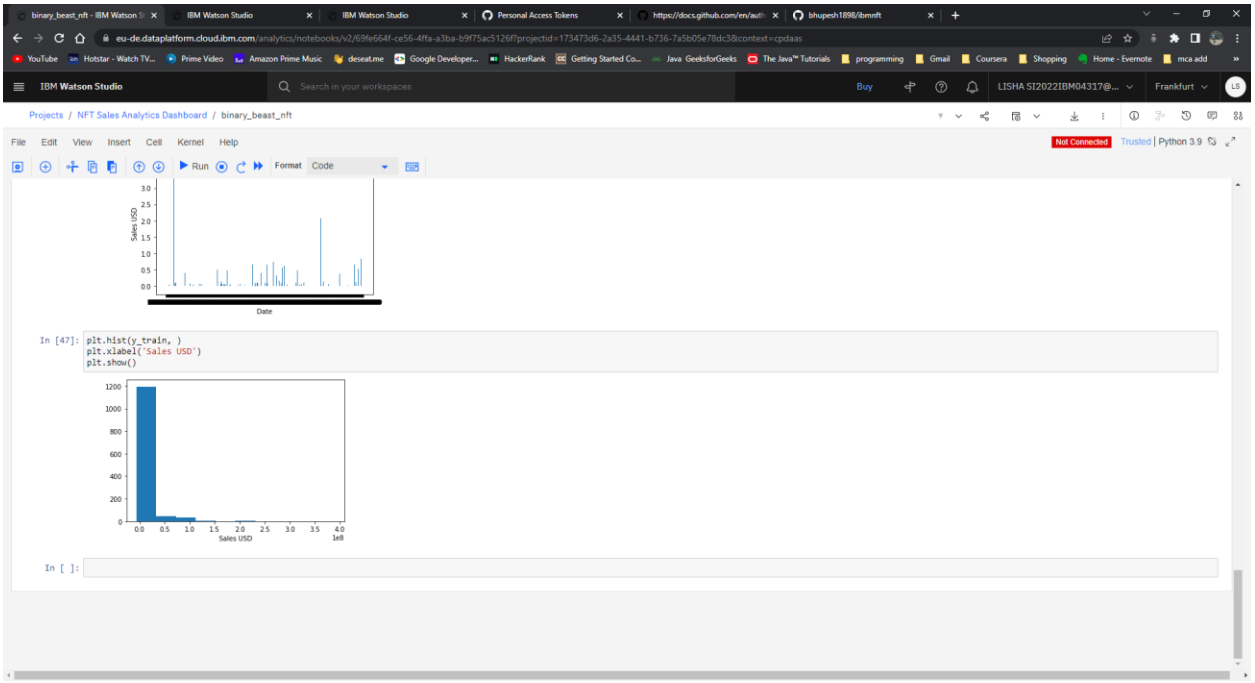
```
In [37]: from sklearn.tree import DecisionTreeRegressor

In [38]: regressor = DecisionTreeRegressor()

In [40]: plt.scatter(x_test, y_test, color = 'red')

         INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
         INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.

Out[40]: <matplotlib.collections.PathCollection at 0x7febf9994be0>

In [41]: df.head()

Out[41]:
```

	Date	Sales_USD_cumsum	Number_of_Sales_cumsum	Active_Market_Wallets_cumsum	Primary_Sales_cumsum	Secondary_Sales_cumsum	AverageUSD_cum	Sales_USD	Number_of_Sales	Active_Market_Wallets	Primary_Sales
0	22-06-2017	0.00	0	0.0	0	NaN	NaN	NaN	NaN	NaN	NaN
1	23-06-2017	1020.30	19	8.0	0	19.0	53.70	1020.30	19.0	8.0	0.0
2	24-06-2017	2261.14	40	21.0	0	21.0	56.53	1240.84	21.0	13.0	0.0
3	25-06-2017	2778.69	53	28.0	0	13.0	52.43	517.55	13.0	7.0	0.0
4	26-06-2017	3203.32	67	34.0	0	14.0	47.81	424.63	14.0	6.0	0.0

```
In [42]: x = df.iloc[:,8:11].values
         y = df.iloc[:,7:8].values
```



```
In [43]: x.shape

Out[43]: (1606, 3)

In [44]: y.shape

Out[44]: (1606, 1)

In [45]: plt.plot(x_train,y_train)

INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.

Out[45]: [<matplotlib.lines.Line2D at 0x7febf956fd90>]

In [46]: plt.bar(x_train,y_train)
         plt.title('Comparison between Sales USD with respect to yearly and monthly performance')
         plt.xlabel('Date')
         plt.ylabel('Sales USD')
         plt.show()

INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
```


binary_beast_nft - IBM Watson St
IBM Watson Studio
IBM Watson Studio
Personal Access Tokens
https://docs.github.com/en/auth
bhupesh1898/ibmnft
eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/69fe664f-ce56-4ffa-a3ba-b9f75ac5126f?projectid=173473d6-2a35-4441-b736-7a5b05e78dc3&context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
LISHA SI2022IBM04317@...
Frankfurt
Projects / NFT Sales Analytics Dashboard / binary_beast_nft
File
Edit
View
Insert
Cell
Kernel
Help
Not Connected
Trusted
Python 3.9
Run
Format
Code
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Sales USD
Date
1200
1000
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e8
Sales USD

```
In [47]: plt.hist(y_train, )
         plt.xlabel('Sales USD')
         plt.show()

In [ ]:
```