Seller Abuse Prevention - Amazon - Medallion Architecture

```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew
from scipy.stats import kurtosis
from scipy.stats import binom
from scipy.stats import poisson
from scipy.stats import norm
from scipy.stats import chi2_contingency
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import
mean_squared_error,r2_score,confusion_matrix,classification_report
from sklearn.cluster import KMeans

# bronze container
listing_raw=pd.read_csv('listings_raw.csv')
reviews_raw=pd.read_csv('reviews_raw.csv')
sellers_raw=pd.read_csv('sellers_raw.csv')
suspicious_activity_raw=pd.read_csv("suspicious_activity_raw.csv")
```

Step1- Problem framing

Stakeholder -We've seen a spike in customer complaints related to unfair seller behavior—fake reviews, price manipulation, and suspicious account patterns. This not only hurts customer trust but also puts our fair sellers at a disadvantage

Step 2 - KPI Identification Abuse related KPI

1. Percentage of sellers flagged for suspicious activity
2. Top 5 most common abuse types
3. Abuse severity distribution
4. Review fraud indicator Rate Listing behavior KPI
5. Top 10 sellers by units sold and revenue Operational KPI
6. Time to detection
7. Percentage of flagged still active

Step - Data collection and cleansing

```
listing_raw.head(n=5)
```

```
  listing_id seller_id product_name     category   price
listed_date  \
0      L0000     S0012         Else     Clothing  394.86         NaN

1      L0001     S0070        Allow     Clothing  394.86  2024-06-28

2      L0002     S0029  Performance        Books  394.86  2024-09-05

3      L0003     S0094    Everybody  Electronics  394.86  2025-05-16

4      L0004     S0036        Offer     Clothing  394.86  2024-09-28


   units_sold
0       248.0
1       437.0
2       772.0
3       739.0
4       657.0
```

```
listing_raw.isna().sum()
```

```
listing_id      5
seller_id       5
product_name    5
category        5
price           5
listed_date     5
units_sold      5
dtype: int64
```

```
listing_raw[listing_raw['listed_date'].isna()]
```

```
   listing_id seller_id product_name     category   price listed_date
\
0       L0000     S0012         Else     Clothing  394.86         NaN

31      L0031     S0059     Increase  Electronics  394.86         NaN

33      L0033     S0082           It         Home  394.86         NaN

62      L0062     S0009     Maintain        Books  394.86         NaN

99      L0099     S0051         Area         Home  394.86         NaN


    units_sold
0        248.0
31       359.0
33       410.0
```

```
62      654.0
99        NaN
```

There are only 5% values are missing, missing values can be dropped but for accuracy im imputing with median date

```
valid_date=listing_raw['listed_date'].dropna()
valid_date=pd.to_datetime(valid_date)
valid_date=valid_date.median()
valid_date=print(valid_date.date())
```

```
2024-11-06
```

```
def date_imputation(value):
    if pd.isna(value):
       return '2024-11-06'
    else:
        return value
```

```
listing_raw['listed_date']=listing_raw['listed_date'].apply(date_imput
ation)
```

```
def na_value (value):
    if pd.isna(value):
        return 0
    else:
        return value
```

```
def na_value_str(value):
    if pd.isna(value):
        return 'unknown'
    else:
        return value
```

```
listing_raw[['listing_id','seller_id','product_name','category']]=list
ing_raw[['listing_id','seller_id','product_name','category']].applymap
(na_value_str)
```

```
listing_raw[['price','units_sold']]=listing_raw[['price','units_sold']
].applymap(na_value)
```

```
listing_raw.isna().sum()  #bingoo
```

```
listing_id      0
seller_id       0
product_name    0
category        0
price           0
listed_date     0
units_sold      0
dtype: int64
```

```python
#changing data types
listing_raw=listing_raw.astype({'category':'category','price':float})

listing_raw['listed_date']=pd.to_datetime(listing_raw['listed_date'])

reviews_raw.head()
```

```
  review_id listing_id review_date  rating  \
0       NaN      L0086  2024-11-02     5.0
1     R0001      L0013         NaN     5.0
2     R0002      L0045  2025-03-11     3.0
3     R0003      L0069  2024-08-31     1.0
4     R0004      L0030  2025-05-02     1.0

                                         review_text  \
0  Dinner Congress citizen off offer even see col...
1                                                NaN
2  Job account like lead door five happy write re...
3     Apply buy sell civil line design early speech.
4    Few break contain its above senior toward part.

                            reviewer_id
0  02ee07fe-ab91-40b1-b138-61c5074b459d
1  47e7beb8-d1a9-46a4-8d9d-fdfc94645271
2  3fbb3ac4-7052-4e63-a33f-af8721feec90
3  5413dc16-3622-42c0-bbad-c5f789276231
4  869d2290-92f0-43b2-b655-1cb0ff7aeadb
```

```python
reviews_raw[['review_id','listing_id','review_text','reviewer_id']]=reviews_raw[['review_id','listing_id','review_text','reviewer_id']].applymap(na_value_str)

reviews_raw['rating']=reviews_raw['rating'].apply(na_value)

reviews_raw['review_date']=reviews_raw['review_date'].apply(date_imputation)

reviews_raw.dtypes
```

```
review_id       object
listing_id      object
review_date     object
rating         float64
review_text     object
reviewer_id     object
dtype: object
```

```python
reviews_raw['review_date']=pd.to_datetime(reviews_raw['review_date'])

reviews_raw.dtypes
```

```
review_id                 object
listing_id                object
review_date       datetime64[ns]
rating                   float64
review_text               object
reviewer_id               object
dtype: object
```

```python
sellers_raw[['seller_id','seller_name','country','email']]=sellers_raw
[['seller_id','seller_name','country','email']].applymap(na_value)

valid_dat1=sellers_raw['account_created'].dropna()

valid_dat1=pd.to_datetime(sellers_raw['account_created'])

print(valid_dat1.date())
```

```
2024-05-23
```

```python
def date_imputation2(value):
    if pd.isna(value):
        return '2024-05-23'
    else:
        return value

sellers_raw['account_created']=sellers_raw['account_created'].apply(da
te_imputation2)

suspicious_activity_raw[['activity_id','seller_id','activity_type','se
verity']]=suspicious_activity_raw[['activity_id','seller_id','activity
_type','severity']].applymap(na_value)

suspicious_activity_raw['detected_on'].dropna(inplace=True)

suspicious_activity_raw['detected_on']=pd.to_datetime(suspicious_activ
ity_raw['detected_on'])
```

“80% of data science is data cleaning. The rest is complaining about cleaning the data.”

```python
#Silver container

listing_raw.to_csv(r"C:\Users\Prem M\Desktop\seller_abuse_raw_dataset\
listing_cleaned.csv", index=False)

sellers_raw.to_csv(r"C:\Users\Prem M\Desktop\seller_abuse_raw_dataset\
sellers_cleaned.csv",index=False)

suspicious_activity_raw.to_csv(r"C:\Users\Prem M\Desktop\
seller_abuse_raw_dataset\suspicious_cleaned.csv",index=False)

#Exploratory data analysis
```

```
df1=pd.read_csv("listing_cleaned.csv")
df2=pd.read_csv("sellers_cleaned.csv")
df3=pd.read_csv("suspicious_cleaned.csv")

merge_df=pd.merge(df1,df2 ,on='seller_id',how='inner')

df=pd.merge(merge_df,df3,on='seller_id',how='inner')

df.tail(n=5)
```

```
   listing_id seller_id product_name     category   price listed_date  \
90      L0095     S0034         Fine         Home  394.86  2024-09-03

91      L0095     S0034         Fine         Home  394.86  2024-09-03

92      L0096     S0025          Gun      Clthing  394.86  2024-12-15

93      L0098     S0093         Make  Electronics  394.86  2025-02-07

94      L0099     S0051         Area         Home  394.86         NaN


    units_sold                 seller_name      account_created country  \
90        61.0                   Henry Inc  2024-02-15 00:00:00      LS
91        61.0                   Henry Inc  2024-02-15 00:00:00      LS
92       217.0                Jones-Tucker  2024-05-09 00:00:00      CO
93       818.0                Roth-Decker  2024-10-27 00:00:00      SM
94         NaN  Montes, Garrison and Davis  2024-12-11 00:00:00      SK

                                  email activity_id       activity_type  \
90                  anthony04@braun.com       A0071  Price Manipulation

91                  anthony04@braun.com       A0084   Multiple Accounts

92                  wholland@gordon.com       A0065        Fake Reviews

93  robertwilson@lindsey-jefferson.com       A0042  Price Manipulation

94                  ptaylor@hotmail.com       A0060  Price Manipulation


   detected_on severity
90  2025-05-16     High
```

```
91  2025-05-16     High
92  2025-05-16      Low
93  2025-05-16      Low
94  2025-05-16      Low
```

```
df.isnull().sum()
```

```
listing_id         0
seller_id          0
product_name       0
category           0
price              0
listed_date        0
units_sold         0
seller_name        0
account_created    0
country            0
email              0
activity_id        0
activity_type      0
detected_on        0
severity           0
dtype: int64
```

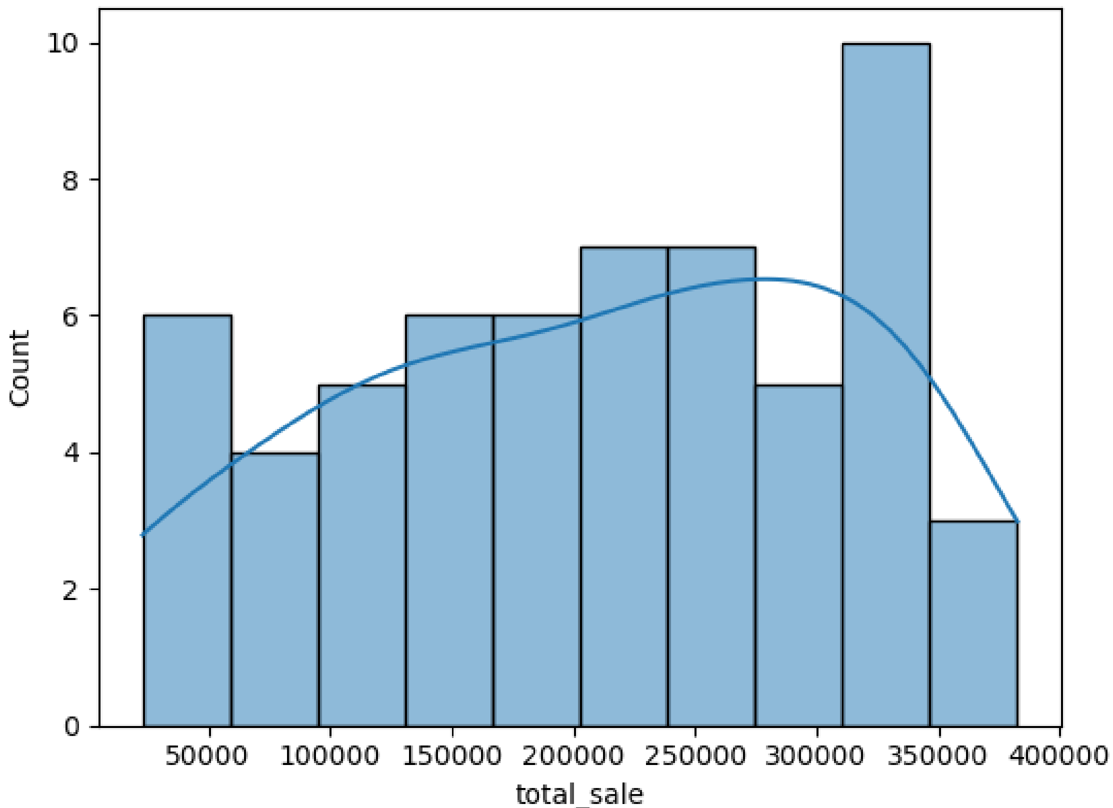
```
df.describe()
```

```
              price  units_sold
count  5.900000e+01   59.000000
mean   3.948600e+02  527.576271
std    3.439881e-13  266.461306
min    3.948600e+02   58.000000
25%    3.948600e+02  323.500000
50%    3.948600e+02  584.000000
75%    3.948600e+02  754.500000
max    3.948600e+02  969.000000
```

```
df['total_sale']=df['price'] * df['units_sold']
```

```
sns.histplot(df['total_sale'],bins=10,kde=True)
```

```
<Axes: xlabel='total_sale', ylabel='Count'>
```




```
skew(df['total_sale'])
```

```
-0.24000616169703287
```

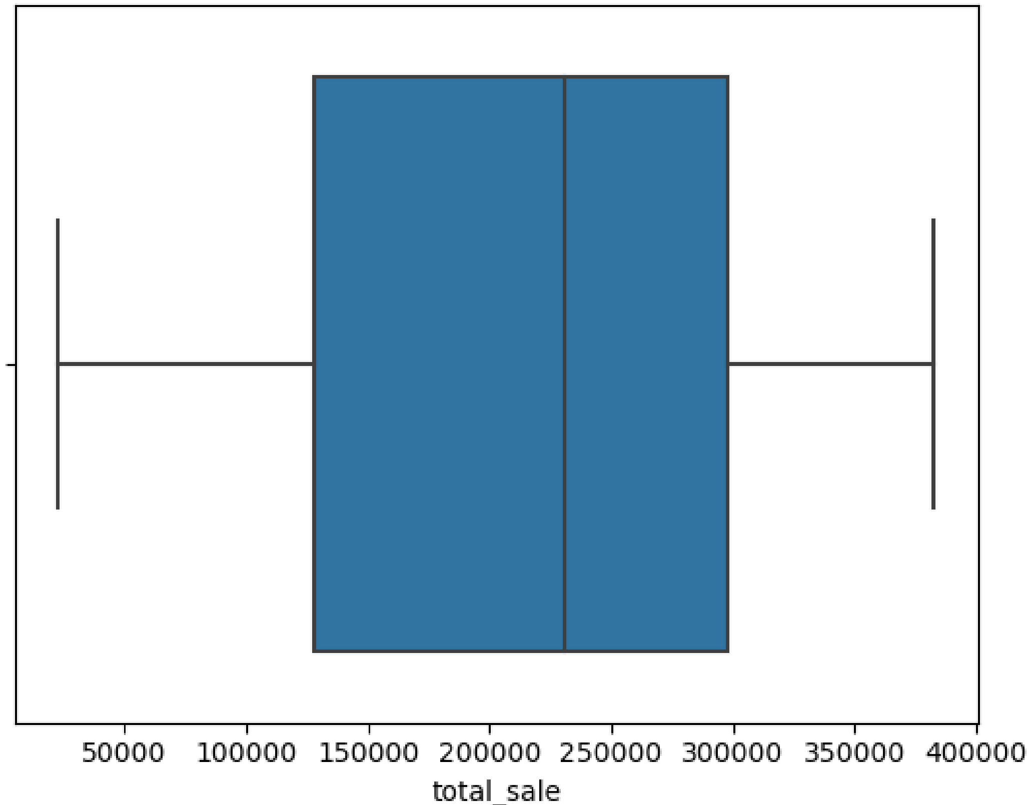
most sellers have similar sales, it is mildly skewed and approximately symmetric

```
kurtosis(df['total_sale'])
```

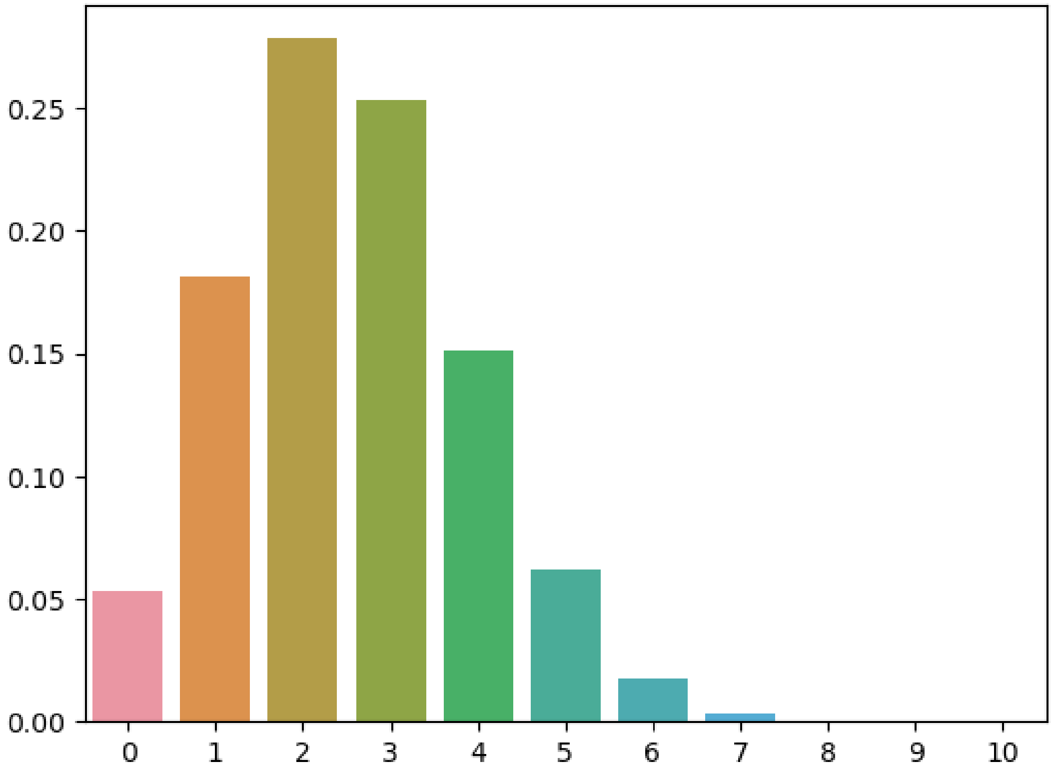
```
-1.0785060796731463
```

No extreme high-performers or underperformers are distorting the data

```
%matplotlib inline
sns.boxplot(data=df,x='total_sale')
plt.show()
```




```
#no outliers

#binomial distribution

high_sales=df['total_sale'] > 300000

n=10
p=np.mean(high_sales)
k=np.arange(0,n+1)

binom_pmf=binom.pmf(k,n,p)

sns.barplot(x=k,y=binom_pmf)

<Axes: >
```



Out of a random sample of 10 sellers, the probability of exactly 2,3 sellers having a total sale greater than ₹3,00,000 is approximately 0.25 (i.e., 25%).

```
df

   listing_id seller_id product_name     category   price listed_date  \
2       L0032     S0012         Time         Toys  394.86  2024-08-17

3       L0003     S0094    Everybody  Electronics  394.86  2025-05-16

4       L0003     S0094    Everybody  Electronics  394.86  2025-05-16

5       L0003     S0094    Everybody  Electronics  394.86  2025-05-16

9       L0004     S0036        Offer     Clothing  394.86  2024-09-28

10      L0093     S0036       Myself  Electronics  394.86  2025-05-14

13      L0027     S0095          Fly        Books  394.86  2025-04-04

14      L0027     S0095          Fly        Books  394.86  2025-04-04

15      L0072     S0095         Idea     Clothing  394.86  2025-05-03
```

16	L0072	S0095	Idea	Clothing	394.86	2025-05-03
19	L0090	S0095	Per	Home	394.86	2024-07-05
20	L0090	S0095	Per	Home	394.86	2024-07-05
21	L0092	S0095	Develop	Home	394.86	2025-01-08
22	L0092	S0095	Develop	Home	394.86	2025-01-08
23	L0011	S0073	Lead	Electronics	394.86	2024-06-19
25	L0012	S0009	Above	Clothing	394.86	2024-09-25
30	L0025	S0088	Story	Electronics	394.86	2024-10-25
31	L0025	S0088	Story	Electronics	394.86	2024-10-25
33	L0017	S0026	Now	Clothing	394.86	2024-11-06
34	L0019	S0032	Recognize	Toys	394.86	2025-02-14
36	L0020	S0052	Provide	Books	394.86	2024-10-31
37	L0020	S0052	Provide	Books	394.86	2024-10-31
39	L0046	S0071	Simple	eletronics	394.86	2025-01-31
40	L0070	S0071	Want	Clthing	394.86	2024-05-20
41	L0097	S0071	Our	Books	394.86	2025-04-18
44	L0024	S0075	Special	Clothing	394.86	2025-01-25
45	L0028	S0045	East	Toyz	394.86	2025-01-02
46	L0028	S0045	East	Toyz	394.86	2025-01-02
47	L0030	S0053	Sit	Toys	394.86	2025-03-06
48	L0030	S0053	Sit	Toys	394.86	2025-03-06
50	L0030	S0053	Sit	Toys	394.86	2025-03-06
51	L0030	S0053	Sit	Toys	394.86	2025-03-06
55	L0038	S0091	Factor	Books	394.86	2025-05-14
58	L0054	S0005	Great	Books	394.86	2024-06-14
60	L0047	S0010	Option	Clothing	394.86	2024-08-31

```
64      L0059     S0041        Figure        Books  394.86  2025-05-16
65      L0059     S0041        Figure        Books  394.86  2025-05-16
66      L0059     S0041        Figure        Books  394.86  2025-05-16
67      L0060     S0055         Woman  Electronics  394.86  2024-12-14
69      L0061     S0047         Fight  Electronics  394.86  2025-05-05
70      L0061     S0047         Fight  Electronics  394.86  2025-05-05
71      L0063     S0011      Business         Home  394.86  2024-06-23
72      L0065     S0017          Join         Toys  394.86  2025-01-24
73      L0068     S0069          Care  Electronics  394.86  2024-09-02
74      L0068     S0069          Care  Electronics  394.86  2024-09-02
77      L0071     S0063         Night  Electronics  394.86  2024-12-19
79      L0076     S0002         Exist        Books  394.86  2024-10-11
80      L0077     S0086        Behind        Books  394.86  2024-12-22
81      L0089     S0086           Six         Toys  394.86  2024-10-03
82      L0080     S0085    Management  Electronics  394.86  2024-07-08
83      L0080     S0085    Management  Electronics  394.86  2024-07-08
84      L0086     S0050         Write  Electronics  394.86  2024-11-04
87      L0094     S0016           Its         Home  394.86  2024-12-12
88      L0094     S0016           Its         Home  394.86  2024-12-12
89      L0095     S0034          Fine         Home  394.86  2024-09-03
90      L0095     S0034          Fine         Home  394.86  2024-09-03
91      L0095     S0034          Fine         Home  394.86  2024-09-03
92      L0096     S0025           Gun      Clthing  394.86  2024-12-15
93      L0098     S0093          Make  Electronics  394.86  2025-02-07

    units_sold                      seller_name account_created
country  \
2        542.0                 Mcdaniel-Cabrera      2023-11-19
```

```
HN
3          739.0                  Cook-Reynolds      2023-09-27
VU
4          739.0                  Cook-Reynolds      2023-09-27
VU
5          739.0                  Cook-Reynolds      2023-09-27
VU
9          657.0      Watkins, Martinez and Russo      2024-07-10
PY
10         338.0      Watkins, Martinez and Russo      2024-07-10
PY
13         200.0                              0      2025-04-29
FJ
14         200.0                              0      2025-04-29
FJ
15         869.0                              0      2025-04-29
FJ
16         869.0                              0      2025-04-29
FJ
19         868.0                              0      2025-04-29
FJ
20         868.0                              0      2025-04-29
FJ
21         274.0                              0      2025-04-29
FJ
22         274.0                              0      2025-04-29
FJ
23         903.0                 Reyes-Campbell      2023-08-23
GE
25         907.0                    Lee-Watkins      2024-12-12
JO
30         342.0                   Short-Moreno      2023-11-21
JP
31         342.0                   Short-Moreno      2023-11-21
JP
33          69.0                   Brewer-Brady      2023-12-16
0
34         360.0                  Thompson LLC      2024-03-27
CF
36         770.0                       Barr Inc      2024-03-05
MN
37         770.0                       Barr Inc      2024-03-05
MN
39         677.0               Williams-Serrano      2024-05-23
ID
40         201.0               Williams-Serrano      2024-05-23
ID
41         263.0               Williams-Serrano      2024-05-23
ID
```

44	498.0	Williams, Davis and Anderson	2025-02-25	NR
45	815.0	Doyle LLC	2024-04-26	ES
46	815.0	Doyle LLC	2024-04-26	ES
47	584.0	Mendoza, Velez and Boyd	2024-07-22	MX
48	584.0	Mendoza, Velez and Boyd	2024-07-22	MX
50	584.0	Mendoza, Velez and Boyd	2024-07-22	MX
51	584.0	Mendoza, Velez and Boyd	2024-07-22	MX
55	563.0	Salinas, Irwin and Lewis	2024-06-28	MM
58	432.0	Clark PLC	2024-06-04	GD
60	874.0	Miller, Richardson and Parker	2024-05-23	AR
64	444.0	Green Group	2024-09-19	EG
65	444.0	Green Group	2024-09-19	EG
66	444.0	Green Group	2024-09-19	EG
67	624.0	Reeves, Stokes and Jordan	2024-04-07	BH
69	58.0	Adams Group	2025-04-23	GA
70	58.0	Adams Group	2025-04-23	GA
71	256.0	Hill and Sons	2023-06-06	TG
72	466.0	Luna-Medina	2024-09-18	GW
73	418.0	Phillips, Rodriguez and Rodgers	2024-08-20	CU
74	418.0	Phillips, Rodriguez and Rodgers	2024-08-20	CU
77	597.0	Avila LLC	2024-07-14	SY
79	969.0	Davis-Owens	2024-01-23	ID
80	661.0	Edwards PLC	2023-11-28	RO
81	620.0	Edwards PLC	2023-11-28	RO
82	678.0	Hernandez-Black	2024-01-03	

```
FM
83      678.0                  Hernandez-Black      2024-01-03
FM
84      309.0                                0      2023-11-18
SA
87      827.0                          Fox Inc      2023-10-20
NE
88      827.0                          Fox Inc      2023-10-20
NE
89       61.0                        Henry Inc      2024-02-15
LS
90       61.0                        Henry Inc      2024-02-15
LS
91       61.0                        Henry Inc      2024-02-15
LS
92      217.0                     Jones-Tucker      2024-05-09
CO
93      818.0                      Roth-Decker      2024-10-27
SM

                                    email activity_id       activity_type  \
2               ellisjustin@gmail.com       A0021    Policy Violation

3             travisdavis@hotmail.com       A0022    Policy Violation

4             travisdavis@hotmail.com       A0029       Pricing Manip

5             travisdavis@hotmail.com       A0075        Fake Reviews

9                   jacob77@gmail.com       A0041   Multiple Accounts

10                  jacob77@gmail.com       A0041   Multiple Accounts

13           lewisraymond@hubbard.net       A0016        Fake Reviews

14           lewisraymond@hubbard.net       A0082        Fake Reviews

15           lewisraymond@hubbard.net       A0016        Fake Reviews

16           lewisraymond@hubbard.net       A0082        Fake Reviews

19           lewisraymond@hubbard.net       A0016        Fake Reviews

20           lewisraymond@hubbard.net       A0082        Fake Reviews

21           lewisraymond@hubbard.net       A0016        Fake Reviews

22           lewisraymond@hubbard.net       A0082        Fake Reviews

23                     ashaw@mayo.org       A0012   Multiple Accounts
```

25	cameron71@kennedy.com	A0033	Fake Reviews
30	nicoleanderson@hotmail.com	A0020	Pricing Manip
31	nicoleanderson@hotmail.com	A0052	Policy Violation
33	zsuarez@hotmail.com	0	FakeRevws
34	xrose@wang-ramirez.org	A0028	Price Manipulation
36	mccannkaren@collins-duran.info	A0009	Fake Reviews
37	mccannkaren@collins-duran.info	A0044	Price Manipulation
39	pamela54@gmail.com	A0006	Fake Reviews
40	pamela54@gmail.com	A0006	Fake Reviews
41	pamela54@gmail.com	A0006	Fake Reviews
44	rodney72@yahoo.com	A0070	Multiple Accounts
45	malonebarry@yahoo.com	A0025	Policy Violation
46	malonebarry@yahoo.com	0	Fake Reviews
47	0	A0017	Policy Violation
48	0	A0027	Policy Violation
50	0	A0058	Multiple Accounts
51	0	A0081	Price Manipulation
55	howellmelissa@mejia.com	A0051	Price Manipulation
58	0	A0049	Policy Violation
60	ethompson@hotmail.com	A0004	Policy Violation
64	patriciabowman@gmail.com	A0040	Price Manipulation
65	patriciabowman@gmail.com	A0047	Fake Reviews
66	patriciabowman@gmail.com	A0096	Price Manipulation
67	richardlopez@yahoo.com	A0074	Mult Acc
69	yorkmichael@gmail.com	A0000	Price Manipulation
70	yorkmichael@gmail.com	A0061	0

71	aaron14@christian-richardson.net	A0087	Price Manipulation
72	opratt@hotmail.com	A0002	Policy Violation
73	milesluis@martin.biz	A0057	Price Manipulation
74	milesluis@martin.biz	A0091	Policy Violation
77	nthomas@allen.com	A0078	Price Manipulation
79	haleyjones@gmail.com	0	Price Manipulation
80	dwashington@yahoo.com	A0035	0
81	dwashington@yahoo.com	A0035	0
82	jenniferwalker@scott.com	A0030	Policy Violation
83	jenniferwalker@scott.com	A0080	Policy Violation
84	qaguilar@hotmail.com	A0014	Fake Reviews
87	anna34@jones.org	A0031	Fake Reviews
88	anna34@jones.org	A0085	Fake Reviews
89	anthony04@braun.com	A0018	Pricing Manip
90	anthony04@braun.com	A0071	Price Manipulation
91	anthony04@braun.com	A0084	Multiple Accounts
92	wholland@gordon.com	A0065	Fake Reviews
93	robertwilson@lindsey-jefferson.com	A0042	Price Manipulation

	detected_on	severity	total_sale	month	day
2	2025-05-16	High	214014.12	Friday	Sunday
3	2025-05-16	Low	291801.54	Friday	Wednesday
4	2025-05-16	Low	291801.54	Friday	Wednesday
5	2025-05-16	Low	291801.54	Friday	Wednesday
9	2025-05-16	Low	259423.02	Friday	Wednesday
10	2025-05-16	Low	133462.68	Friday	Wednesday
13	2025-05-16	High	78972.00	Friday	Tuesday
14	2025-05-16	0	78972.00	Friday	Tuesday
15	2025-05-16	High	343133.34	Friday	Tuesday
16	2025-05-16	0	343133.34	Friday	Tuesday
19	2025-05-16	High	342738.48	Friday	Tuesday
20	2025-05-16	0	342738.48	Friday	Tuesday

```
21  2025-05-16     High   108191.64  Friday    Tuesday
22  2025-05-16        0   108191.64  Friday    Tuesday
23  2025-05-16   Medium   356558.58  Friday  Wednesday
25  2025-05-16      Low   358138.02  Friday   Thursday
30  2025-05-16      Low   135042.12  Friday    Tuesday
31  2025-05-16      Low   135042.12  Friday    Tuesday
33  2025-05-16   Medium    27245.34  Friday   Saturday
34  2025-05-16      Low   142149.60  Friday  Wednesday
36  2025-05-16   Medium   304042.20  Friday    Tuesday
37  2025-05-16      Low   304042.20  Friday    Tuesday
39  2025-05-16   Medium   267320.22  Friday   Thursday
40  2025-05-16   Medium    79366.86  Friday   Thursday
41  2025-05-16   Medium   103848.18  Friday   Thursday
44  2025-05-16      Low   196640.28  Friday    Tuesday
45  2025-05-16     High   321810.90  Friday     Friday
46  2025-05-16     High   321810.90  Friday     Friday
47  2025-05-16      Low   230598.24  Friday     Monday
48  2025-05-16   Medium   230598.24  Friday     Monday
50  2025-05-16   Medium   230598.24  Friday     Monday
51  2025-05-16     High   230598.24  Friday     Monday
55  2025-05-16      Low   222306.18  Friday     Friday
58  2025-05-16   Medium   170579.52  Friday    Tuesday
60  2025-05-16   Medium   345107.64  Friday   Thursday
64  2025-05-16   Medium   175317.84  Friday   Thursday
65  2025-05-16   Medium   175317.84  Friday   Thursday
66  2025-05-16      Low   175317.84  Friday   Thursday
67  2025-05-16   Medium   246392.64  Friday     Sunday
69  2025-05-16     High    22901.88  Friday  Wednesday
70  2025-05-16   Medium    22901.88  Friday  Wednesday
71  2025-05-16     High   101084.16  Friday    Tuesday
72  2025-05-16   Medium   184004.76  Friday  Wednesday
73  2025-05-16        0   165051.48  Friday    Tuesday
74  2025-05-16   Medium   165051.48  Friday    Tuesday
77  2025-05-16   Medium   235731.42  Friday     Sunday
79  2025-05-16     High   382619.34  Friday    Tuesday
80  2025-05-16     High   261002.46  Friday    Tuesday
81  2025-05-16     High   244813.20  Friday    Tuesday
82  2025-05-16     High   267715.08  Friday  Wednesday
83  2025-05-16      Low   267715.08  Friday  Wednesday
84  2025-05-16   Medium   122011.74  Friday   Saturday
87  2025-05-16      Low   326549.22  Friday     Friday
88  2025-05-16      Low   326549.22  Friday     Friday
89  2025-05-16      Low    24086.46  Friday   Thursday
90  2025-05-16     High    24086.46  Friday   Thursday
91  2025-05-16     High    24086.46  Friday   Thursday
92  2025-05-16      Low    85684.62  Friday   Thursday
93  2025-05-16      Low   322995.48  Friday     Sunday
```

```
#chi square test
```

```
contingency_table=pd.crosstab(df['category'],df['activity_type'])

chi_tes,p_value,dof,expected_value=chi2_contingency(contingency_table)

if p_value < 0.05:
    print("reject null hypothesis there is a difference between 
category and activity type")
else:
    print("failed_to_reject_null hypothesis, there is no difference 
between them")
```

```
failed_to_reject_null hypothesis, there is no difference between them
```

```
#machine learning

model=LinearRegression()

x=df[['units_sold']]
y=df['total_sale']

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,rando
m_state=42)

model.fit(x_train,y_train)
```

```
LinearRegression()
```

```
y_predicted=model.predict(x_test)

pd.DataFrame({'units_sold':x_test.values.flatten(),'total_sale':y_test
.values.flatten(),'predicted_sale':y_predicted})
```

	units_sold	total_sale	predicted_sale
0	542.0	214014.12	214014.12
1	338.0	133462.68	133462.68
2	874.0	345107.64	345107.64
3	274.0	108191.64	108191.64
4	597.0	235731.42	235731.42
5	827.0	326549.22	326549.22
6	217.0	85684.62	85684.62
7	498.0	196640.28	196640.28
8	661.0	261002.46	261002.46
9	274.0	108191.64	108191.64
10	678.0	267715.08	267715.08
11	739.0	291801.54	291801.54

```
 mean_squared_error(y_test,y_predicted) #bingo accurate prediction
```

```
9.529120656610879e-22
```

```
values=np.array([[2100],[2200]])

model.predict(values)
```

```
C:\Users\Prem M\anaconda3\envs\pandas_playground\Lib\site-packages\
sklearn\base.py:464: UserWarning: X does not have valid feature names,
but LinearRegression was fitted with feature names
  warnings.warn(
```

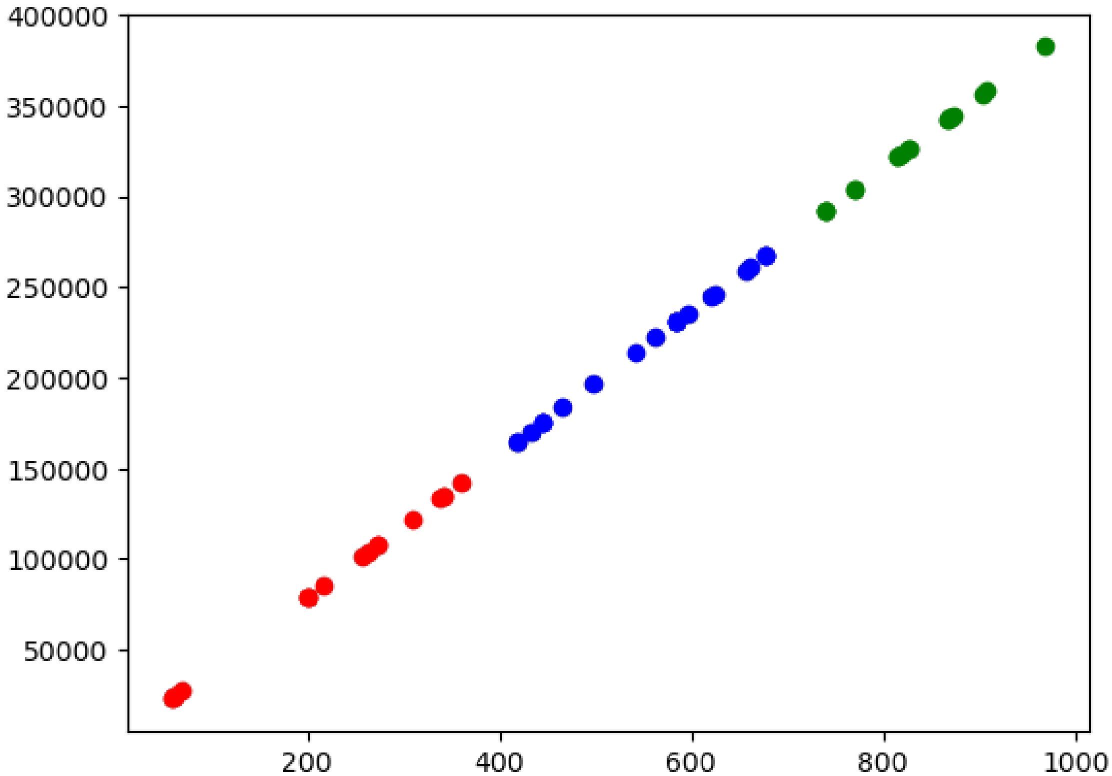
```
array([829206., 868692.])
```

```python
#kmeans

kmeans=KMeans(n_clusters=3,random_state=42)

df['cluster']=kmeans.fit_predict(df[['total_sale']])
```

```
C:\Users\Prem M\anaconda3\envs\pandas_playground\Lib\site-packages\
sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of
`n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Prem M\anaconda3\envs\pandas_playground\Lib\site-packages\
sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have
a memory leak on Windows with MKL, when there are less chunks than
available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```python
df1=df[df['cluster']==0]
df2=df[df['cluster']==1]
df3=df[df['cluster']==2]

plt.scatter(df1['units_sold'],df1['total_sale'],color='blue')
plt.scatter(df2['units_sold'],df2['total_sale'],color='red')
plt.scatter(df3['units_sold'],df3['total_sale'],color='green')
```

```
<matplotlib.collections.PathCollection at 0x1b7e6f5f310>
```


400000
350000
300000
250000
200000
150000
100000
50000
200
400
600
800
1000