```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import KNNImputer
from sklearn import linear_model
from sklearn.metrics import r2_score
from sklearn.tree import DecisionTreeRegressor
from lightgbm import LGBMRegressor from
xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error,r2_score,
confusion_matrix
from sklearn.model_selection import cross_val_score

import warnings
warnings.filterwarnings('ignore')

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146:
UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this
version of SciPy (detected version 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
```

```
/kaggle/input/flipkart-product-dataset/train.csv
/kaggle/input/flipkart-product-dataset/test.csv
```

```
train_df =
pd.read_csv('/kaggle/input/flipkart-product-dataset/train.csv')
test_df =
pd.read_csv('/kaggle/input/flipkart-product-dataset/test.csv')
```

```
train_df.head()
```

```
      id                                               title Rating
maincateg  \
0  16695    Fashionable & Comfortable Bellies For Women  (...     3.9
Women
1   5120  Combo Pack of 4 Casual Shoes Sneakers For Men ...3.8
Men
2   18391        Cilia Mode Leo Sneakers For Women   (White)     4.4
Women
3     495                             Men Black Sports Sandal     4.2
```

```
Men
4  16408                                 Men Green Sports Sandal      3.9
Men

   platform   price1  actprice1 Offer %   norating1   noreviews1  star_5f
\
0  Flipkart      698        999   30.13%       38.0          7.0     17.0

1  Flipkart      999       1999   50.03%      531.0         69.0    264.0

2  Flipkart     2749       4999   45.01%       17.0          4.0     11.0

3  Flipkart      518        724   15.85%    46413.0       6229.0   1045.0

4  Flipkart     1379       2299   40.02%       77.0          3.0     35.0


   star_4f   star_3f  star_2f   star_1f  fulfilled1
0      9.0       6.0        3         3           0
1     92.0      73.0       29        73           1
2      3.0       2.0        1         0           1
3  12416.0    5352.0      701      4595           1
4     21.0       7.0        7         7           1
```

```
train_df.shape
```

```
(15730, 16)
```

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15730 entries, 0 to 15729
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          15730 non-null  int64
 1   title       15730 non-null  object
 2   Rating      15730 non-null  float64
 3   maincateg   15204 non-null  object
 4   platform    15730 non-null  object
 5   price1      15730 non-null  int64
 6   actprice1   15730 non-null  int64
 7   Offer %15730 non-null  object
 8   norating1   15052 non-null  float64
 9   noreviews1  15152 non-null  float64
 10  star_5f15142 non-null  float64
 11  star_4f15191 non-null  float64
 12  star_3f15499 non-null  float64
 13  star_2f15730 non-null  int64
 14  star_1f15730 non-null  int64
 15  fulfilled1  15730 non-null  int64
```

```
dtypes: float64(6), int64(6), object(4)
memory usage: 1.9+ MB
```

```
train_df.describe()
```

```
                   id        Rating         price1     actprice1
norating1  \
count   15730.000000 15730.000000   15730.000000 15730.000000
15052.000000
mean    10479.541577      4.012873     688.070693   1369.286777
3057.660776
std      6080.166276      0.298440     649.409586   1240.900227
11846.965689
min         3.000000      0.000000      69.000000     42.000000
1.000000
25%      5212.000000      3.900000     349.000000    699.000000
63.000000
50%     10458.500000      4.000000     474.000000    999.000000
308.000000
75%     15766.750000      4.200000     699.000000   1299.000000
1526.000000
max     20973.000000      5.000000    5998.000000 13499.000000
289973.000000

          noreviews1        star_5f        star_4f       star_3f
star_2f  \
count     15152.000000   15142.000000   15191.000000  15499.000000
15730.000000
mean       423.976307    1585.239466     655.923310    357.260662
155.085188
std       1768.230384    6177.476241    2855.735531   1402.246610
558.650254
min          0.000000       0.000000       0.000000      0.000000
0.000000
25%          9.000000      30.000000      12.000000      7.000000
3.000000
50%         44.000000     150.000000      60.000000     34.000000
17.000000
75%        215.000000     788.000000     300.000000    172.000000
77.000000
max      45448.000000 151193.000000   74037.000000  34978.000000
11705.000000

            star_1f   fulfilled1
count  15730.000000 15730.000000
mean     275.500572     0.601526
std      958.589075     0.489600
min        0.000000     0.000000
25%        6.000000     0.000000
50%       30.000000     1.000000
```

```
75%        140.000000     1.000000
max      18060.000000     1.000000
```

```
train_df.isnull().sum()
```

```
id              0
title           0
Rating          0
maincateg     526
platform        0
price1          0
actprice1       0
Offer %         0
norating1     678
noreviews1    578
star_5f       588
star_4f       539
star_3f       231
star_2f         0
star_1f         0
fulfilled1      0
dtype: int64
```

Before moving on to EDA, we have to do 2 things:

1.Fill the missing values and modify the data, and,

2. Convert the object columns into numerical (not all though).

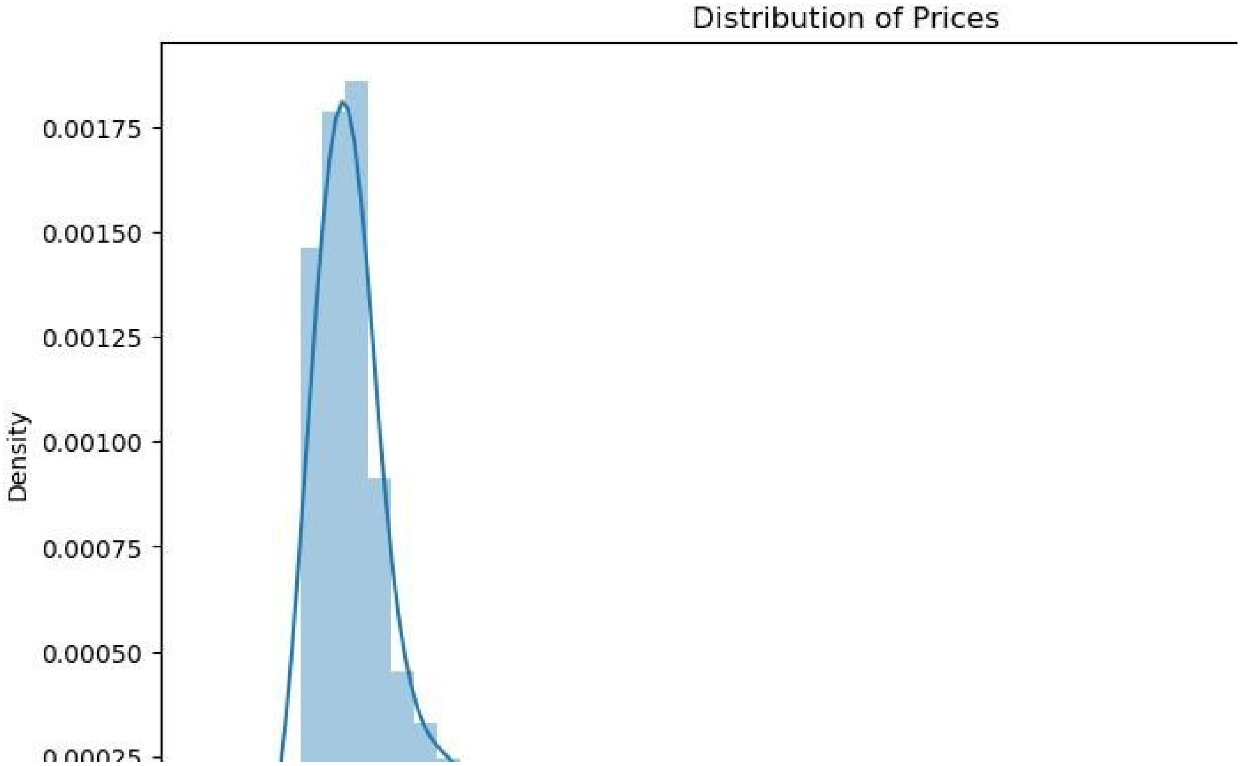
```
# Convert the Offer % column to float
train_df['Offer %'] = train_df['Offer %'].str[:4].astype('float')

# Fill in the missing values in the norating1 and noreviews1 columns
train_df['norating1'] =
train_df['norating1'].fillna(train_df['norating1'].sum())
train_df['noreviews1'] =
train_df['noreviews1'].fillna(train_df['noreviews1'].sum())

# Create a new column called total_cus that sums the values of the
norating1, star_5f, star_4f, star_3f, star_2f, and star_1f columns
total_cus = train_df['norating1'] + train_df['star_5f'] +
train_df['star_4f'] + train_df['star_3f'] + train_df['star_2f'] +
train_df['star_1f']
train_df['total_cus'] = total_cus

# Drop the star_5f, star_4f, star_3f, star_2f, and star_1f columns
train_df = train_df.drop(['star_5f', 'star_4f', 'star_3f', 'star_2f',
'star_1f'], axis=1)

# Drop the norating1 and noreviews1 columns
```

```
train_df = train_df.drop(['norating1', 'noreviews1'], axis=1)

# Convert the platform column to numeric
train_df['platform'] = train_df['platform'].map({'Flipkart': 0,
'Amazon': 1})

# If the maincateg column is null, set it to "Unknown"
train_df.loc[train_df['maincateg'].isnull(), 'maincateg'] = 'Unknown'

# Check if there are any null values in the maincateg column
print(train_df['maincateg'].isnull().sum())

0

print(train_df['maincateg'].unique())

['Women' 'Men' 'Unknown']
```

Time for EDA:

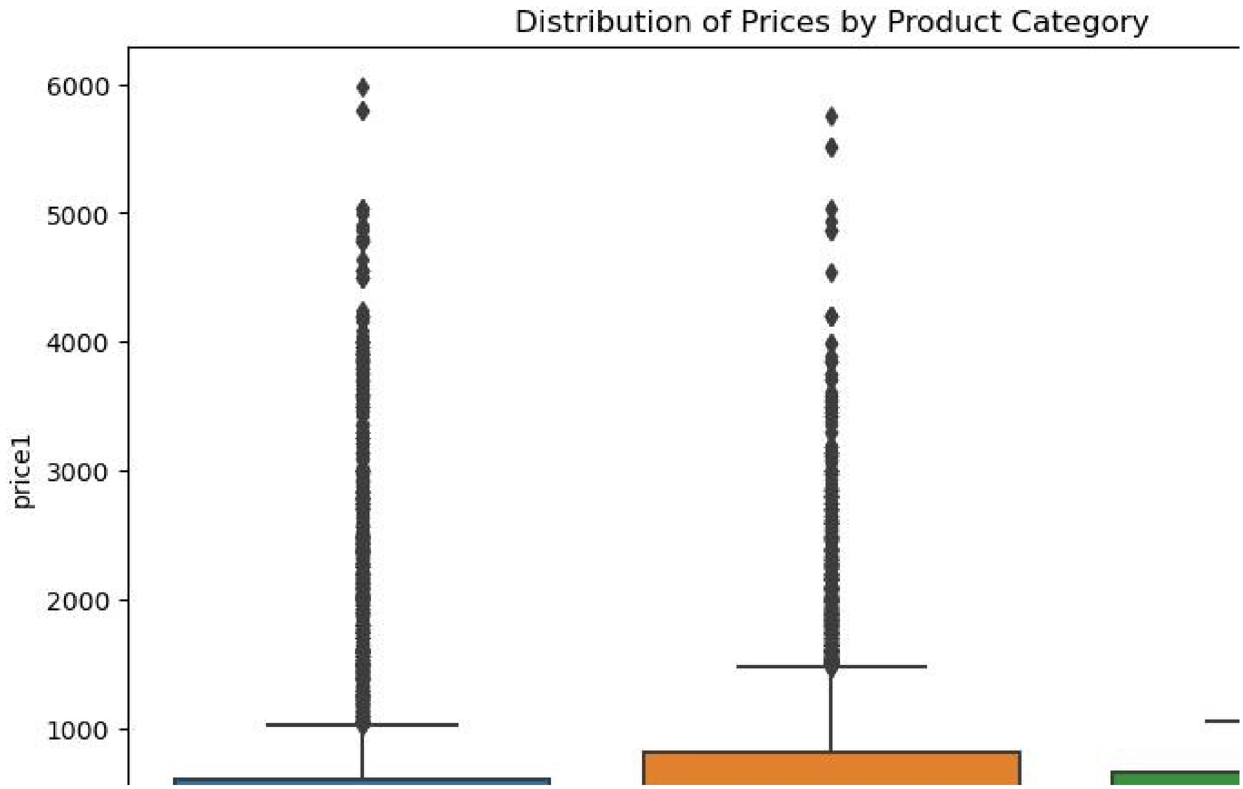
1. Exploring the distribution of prices:

```
plt.figure(figsize=(10, 6))
sns.distplot(train_df['price1'])
plt.title('Distribution of Prices')
plt.show()
```



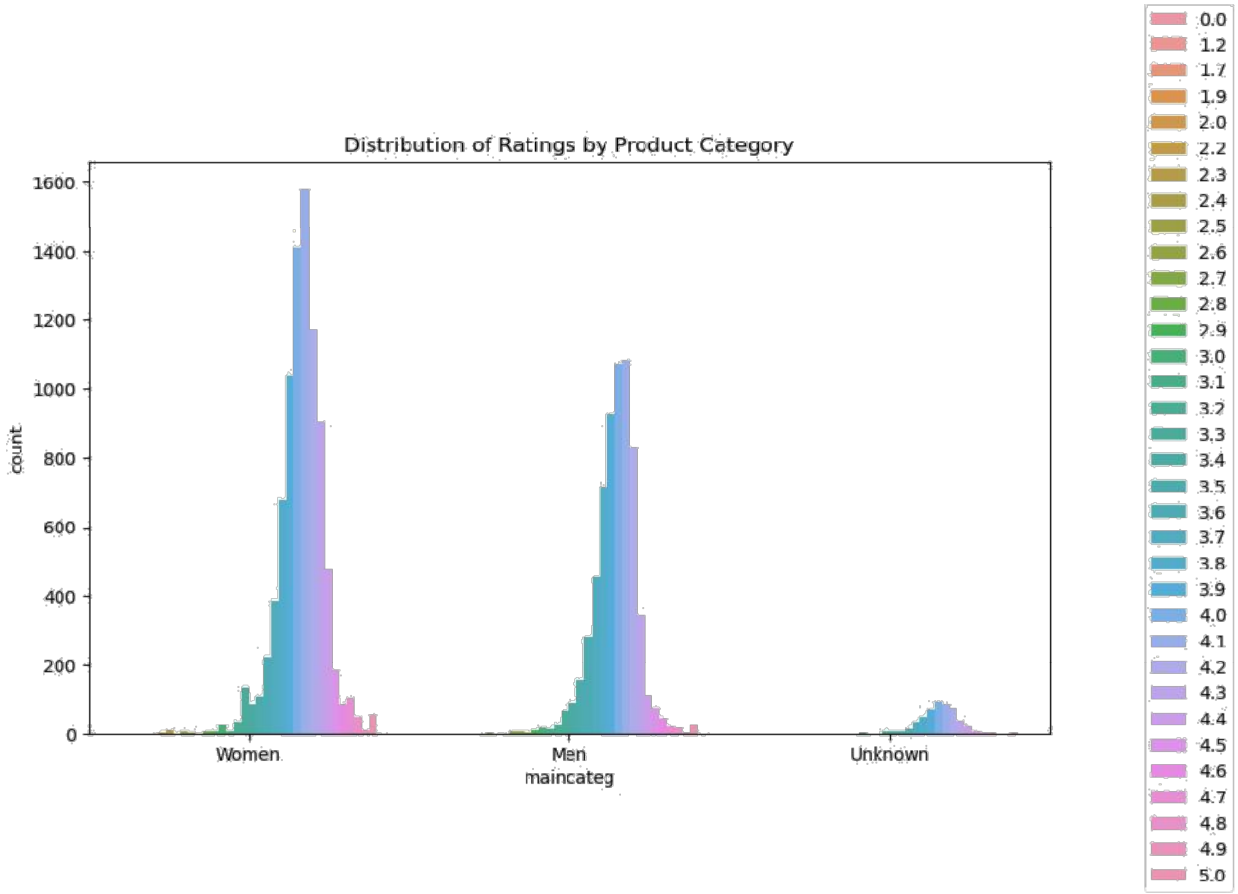
- There are a few products with very high prices, but the majority of products have lower prices.

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='maincateg', y='price1', data=train_df)
plt.title('Distribution of Prices by Product Category')
plt.show()
```



2. Exploring the reviews:

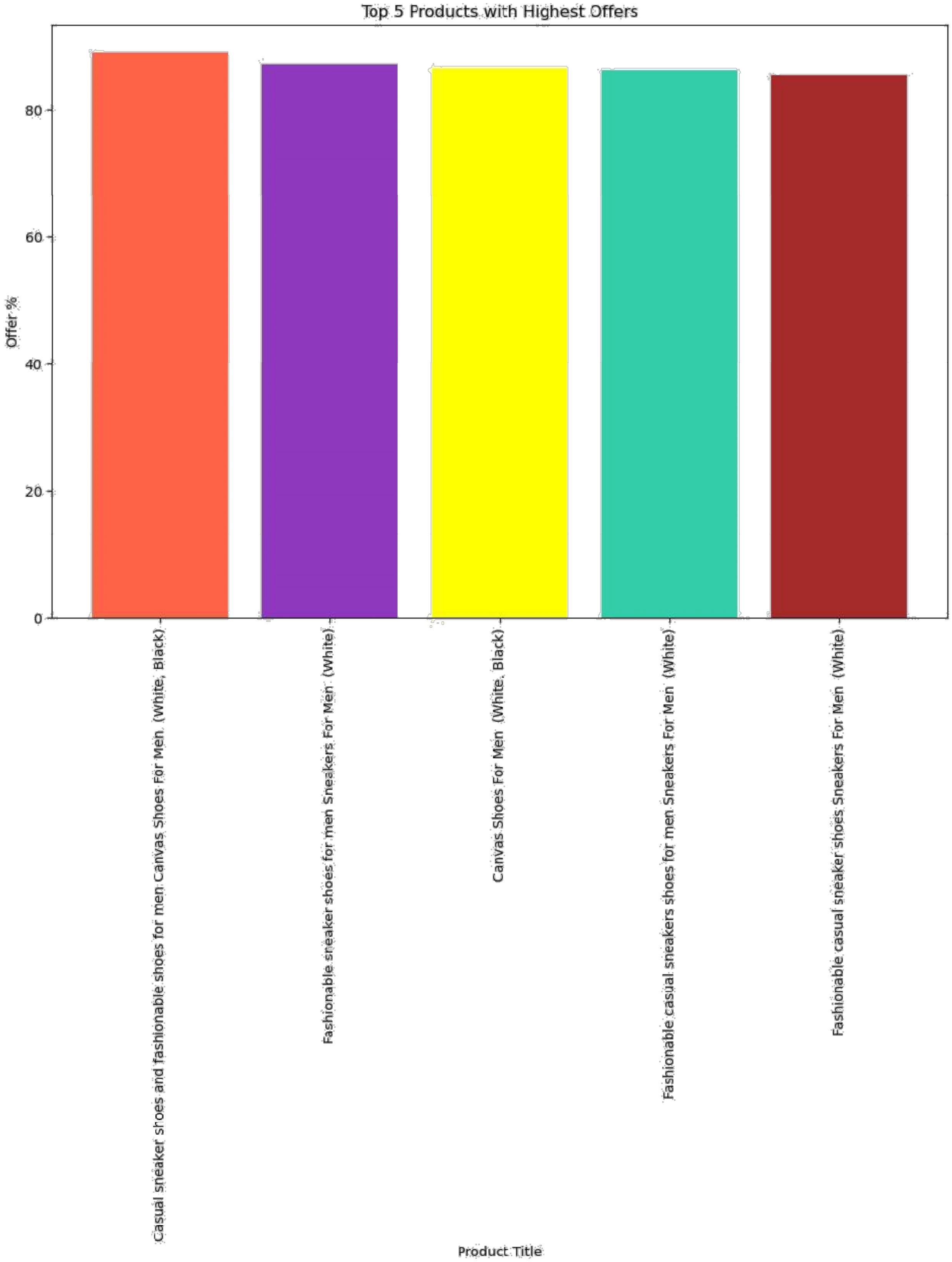
```
plt.figure(figsize=(10, 6))
sns.countplot(x='maincateg', hue='Rating', data=train_df)
plt.title('Distribution of Ratings by Product Category')
plt.legend(bbox_to_anchor=(1.2, 0.5), loc='right')
plt.show()
```



Products in the "Women" category have more "5.0" ratings than the other 2 categories. This may imply that Women shop more! Or other factors also, such as lack of good products in the other categories.

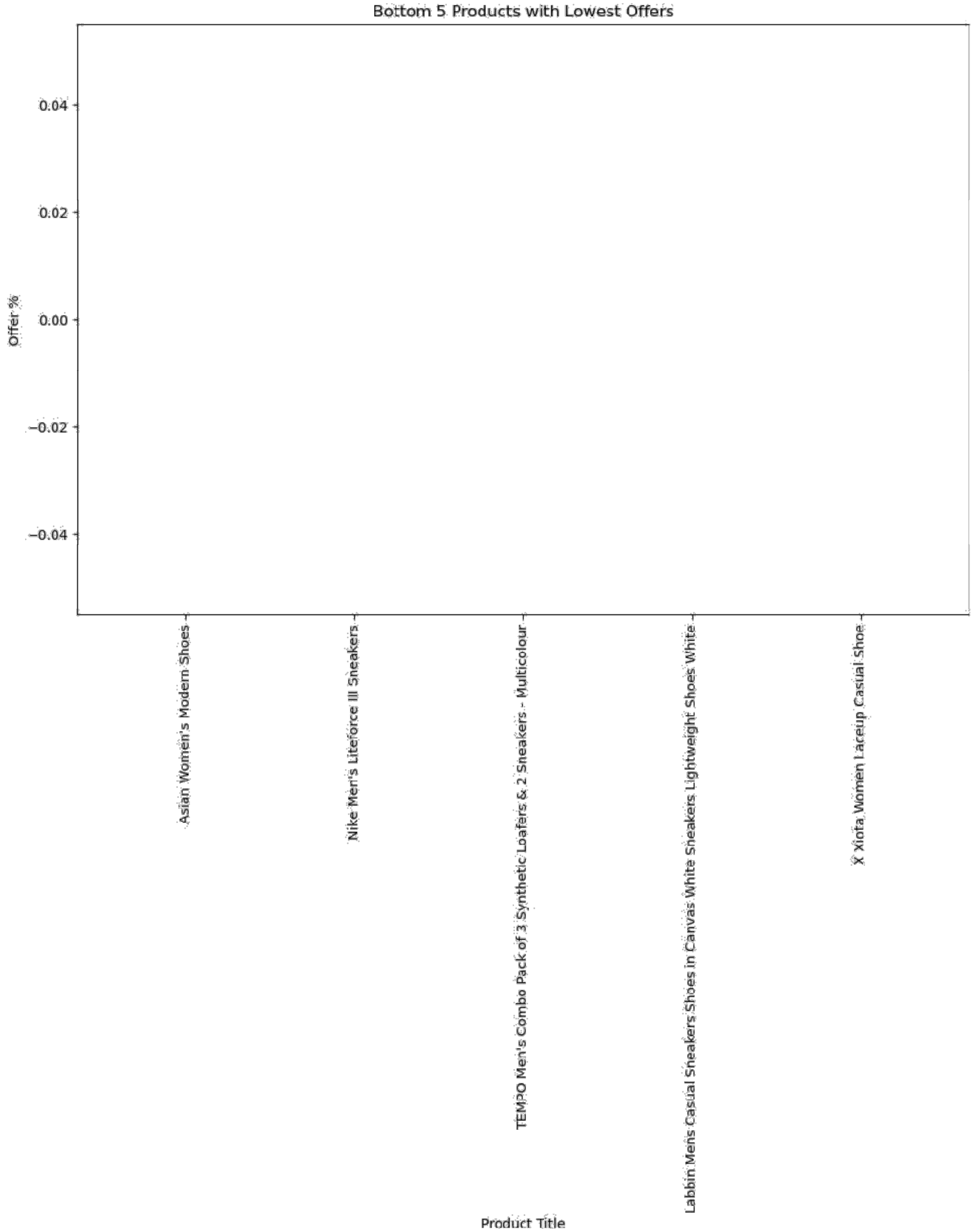
3. Let's see the top 5 products with highest offers:

```
# Get the top 5 product titles with the highest offers
top_5_titles = train_df.groupby('title')['Offer
%'].mean().sort_values(ascending=False).head(5)

# Plot the top 5 product titles with the highest offers
plt.figure(figsize=(12, 8))
plt.bar(top_5_titles.index, top_5_titles.values, color=['#FF6347',
'#8D38BD', '#FFFF00', '#33CDAA', '#A52A2A'])
plt.title('Top 5 Products with Highest Offers')
plt.xlabel('Product Title') plt.ylabel('Offer
%') plt.xticks(rotation=90)

plt.show()
```


Top 5 Products with Highest Offers
Offer %
0
20
40
60
80
Casual sneaker shoes and fashionable shoes for men Canvas Shoes For Men (White, Black)
Fashionable sneaker shoes for men Sneakers For Men (White)
Canvas Shoes For Men (White, Black)
Fashionable casual sneakers shoes for men Sneakers For Men (White)
Fashionable casual sneaker shoes Sneakers For Men (White)
Product Title

Alomost over 85% discounts on the 5 items????? How are the sellers making profits on these?

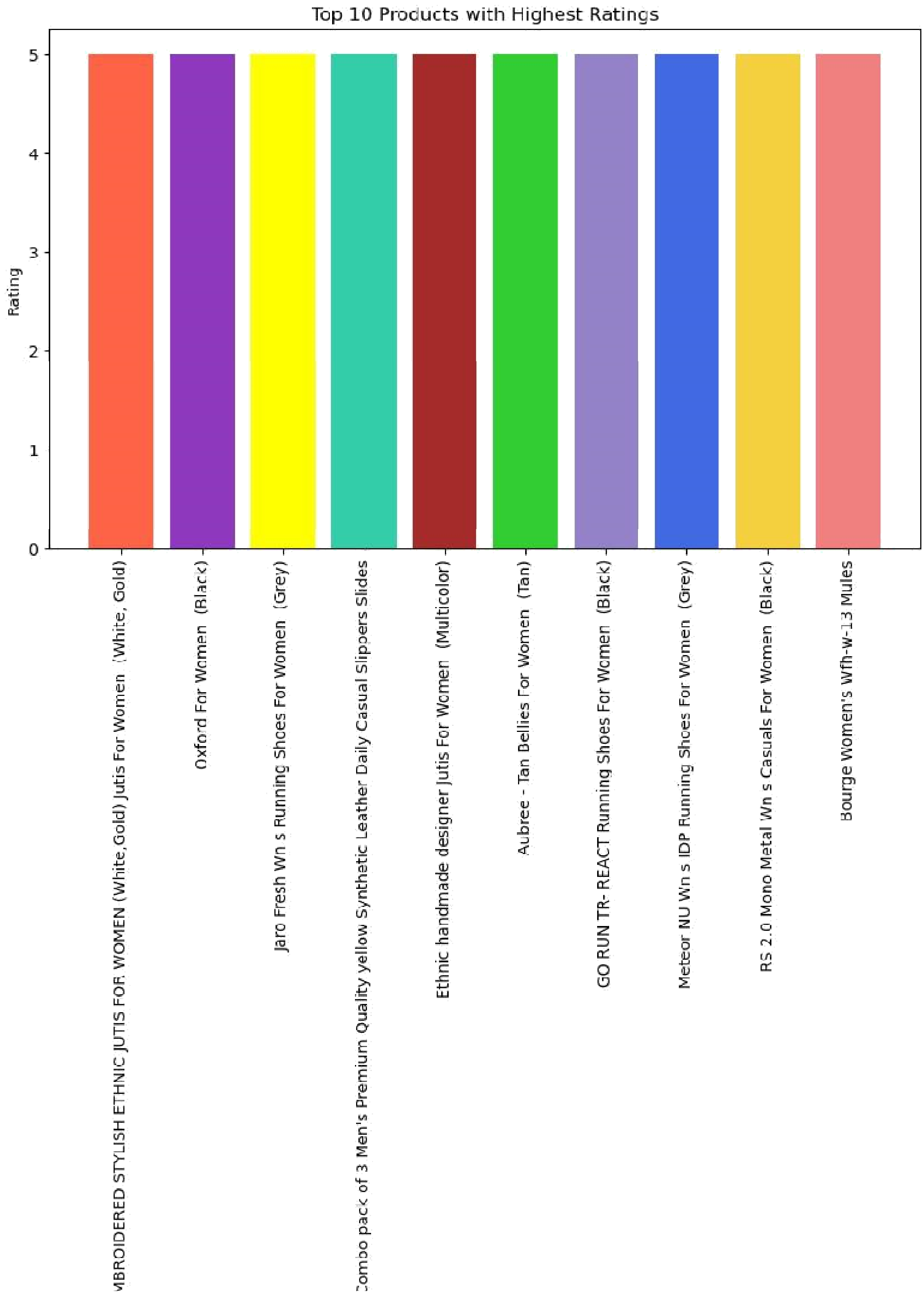
```
# Get the bottom 5 product titles with the lowest offers
bottom_5_titles = train_df.groupby('title')['Offer
%'].mean().sort_values(ascending=True).head(5)

# Plot the bottom 5 product titles with the lowest offers
plt.figure(figsize=(12, 8)) plt.bar(bottom_5_titles.index,
bottom_5_titles.values, color=['#FF6347', '#8D38BD',
'#FFFF00', '#33CDAA', '#A52A2A']) plt.title('Bottom 5 Products
with Lowest Offers') plt.xlabel('Product Title')

plt.ylabel('Offer %')
plt.xticks(rotation=90)
plt.show()
```


Bottom 5 Products with Lowest Offers
Offer %
0.04
0.02
0.00
−0.02
−0.04
Asian Women's Modern Shoes
Nike Men's Liteforce III Sneakers
TEMPO Men's Combo Pack of 3 Synthetic Loafers & 2 Sneakers - Multicolour
Labbin Mens Casual Sneakers Shoes in Canvas White Sneakers Lightweight Shoes White
X Xiota Women Laceup Casual Shoe
Product Title

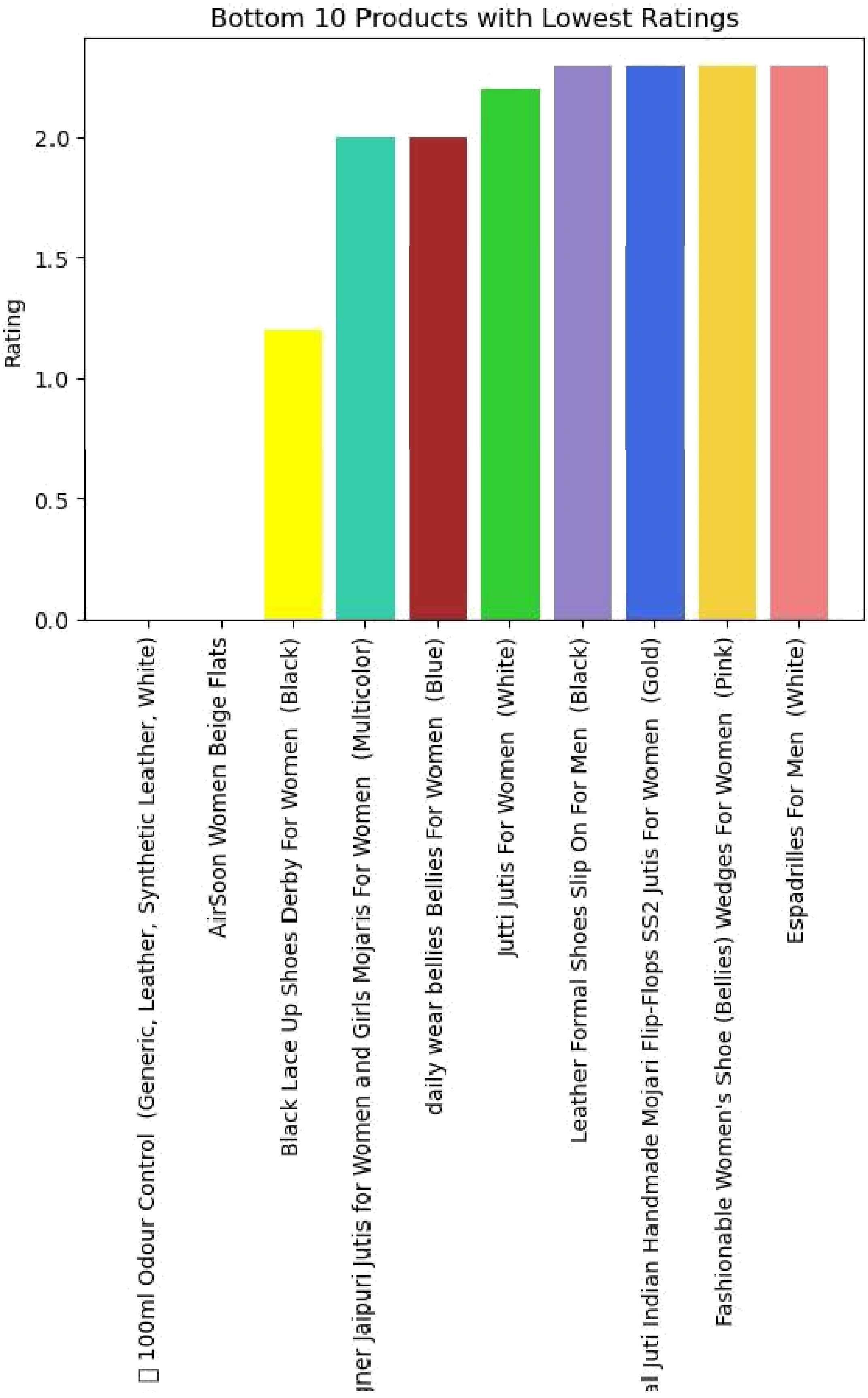
Absolutely no offers on these products.

Let's see products with highest and lowest ratings:

```
# Get the top 10 product titles with the highest ratings
top_10_product_titles = train_df.groupby('title')
['Rating'].mean().sort_values(ascending=False).head(10)

# Plot the top 10 product titles with the highest ratings
plt.figure(figsize=(10, 6)) plt.bar(top_10_product_titles.index,
top_10_product_titles.values, color=['#FF6347', '#8D38BD',
'#FFFF00', '#33CDAA', '#A52A2A', '#32CD32', '#9482C9',


'#4169E1', '#F4D03F', '#F08080'])
plt.title('Top 10 Products with Highest Ratings')
plt.xlabel('Product Title')
plt.xticks(rotation=90)
plt.ylabel('Rating')

plt.show()
```


Top 10 Products with Highest Ratings
Rating
0
1
2
3
4
5
MBROIDERED STYLISH ETHNIC JUTIS FOR WOMEN (White,Gold) Jutis For Women (White, Gold)
Oxford For Women (Black)
Jaro Fresh Wn s Running Shoes For Women (Grey)
Combo pack of 3 Men's Premium Quality yellow Synthetic Leather Daily Casual Slippers Slides
Ethnic handmade designer Jutis For Women (Multicolor)
Aubree - Tan Bellies For Women (Tan)
GO RUN TR- REACT Running Shoes For Women (Black)
Meteor NU Wn s IDP Running Shoes For Women (Grey)
RS 2.0 Mono Metal Wn s Casuals For Women (Black)
Bourge Women's Wfh-w-13 Mules

```
# Get the bottom 10 product titles with the lowest ratings
bottom_10_product_titles = train_df.groupby('title')
['Rating'].mean().sort_values(ascending=True).head(10)

# Plot the bottom 10 product titles with the lowest ratings
plt.bar(bottom_10_product_titles.index,
bottom_10_product_titles.values, color=['#FF6347', '#8D38BD',
'#FFFF00', '#33CDAA', '#A52A2A', '#32CD32', '#9482C9',

'#4169E1', '#F4D03F', '#F08080'])
plt.title('Bottom 10 Products with Lowest Ratings')
plt.xlabel('Product Title')
plt.xticks(rotation=90)
plt.ylabel('Rating')

plt.show()
```


Bottom 10 Products with Lowest Ratings
Rating
0.0
0.5
1.0
1.5
2.0
□ 100ml Odour Control (Generic, Leather, Synthetic Leather, White)
AirSoon Women Beige Flats
Black Lace Up Shoes Derby For Women (Black)
gner Jaipuri Jutis for Women and Girls Mojaris For Women (Multicolor)
daily wear bellies Bellies For Women (Blue)
Jutti Jutis For Women (White)
Leather Formal Shoes Slip On For Men (Black)
al Juti Indian Handmade Mojari Flip-Flops SS2 Jutis For Women (Gold)
Fashionable Women's Shoe (Bellies) Wedges For Women (Pink)
Espadrilles For Men (White)

Moving on to modelling:

```
train_df=pd.get_dummies(train_df,columns=['maincateg'])

from collections import Counter

arr = []
for i in train_df['title'].str.split(' '):
    arr += i

c = Counter(arr)
c_ list= c.most_common( 100)
c_list

# words = []

# for wd in  c_list:
#     words.append(wd[0])

# print(words)
```

```
[('For', 10241),
 ('Sandal', 4194),
 ('Shoes', 4163),
 ('Women', 3976),
 ('Running', 2024),
 ('Sneakers', 1806),
 ('Men', 1580),
 ('Black', 1499),
 ('Boots', 1471),
 ('Casual', 1454),
 ('Men\xa0\xa0(Black)', 1454),
 ('Bellies', 1093),
 ('for', 1080),
 ('Jutis', 1073),
 ('Flats', 1054),
 ('Heels', 1008),
 ('Women\xa0\xa0(Black)', 967),
 ('Flip', 894),
 ('Wedges', 875),
 ('Flops', 843),
 ('Stylish', 830),
 ('Walking', 775),
 ('Slip', 711),
 ('Slippers', 679),
 ('On', 663),
 ('Sports', 653),
 ('&', 639),
 ('Slides', 618),
 ('Casuals', 614),
 ('shoes', 593),
('Girls', 592),
('Women\xa0\xa0(Pink)', 578),
('Men\xa0\xa0(Brown)', 529),
('|', 505),
('Leather', 469),
('of', 458),
('Mojaris', 458),
('Pink', 455),
("Men's", 445),
('Women\xa0\xa0(Multicolor)', 442),
('Lace', 435),
('Brown', 435),
('Women\xa0\xa0(White)', 432),
('and', 430),
('Combo', 403),
('Grey', 378),
('Blue', 371),
('Pack', 361),
('Women\xa0\xa0(Grey)', 360),
('Men\xa0\xa0(Blue)', 356),
('High', 351),
('Latest', 348),
('Loafers', 337),
(',', 335),
('2', 333),
('Men\xa0\xa0(White)', 332),
('Women\xa0\xa0(Blue)', 329),
('Up', 323),
('Beige', 321),
('Comfortable', 316),
("Women's", 316),
('White', 310),
('Red', 309),
('men', 299),
('Men\xa0\xa0(Tan)', 284),
('And', 283),
('Ethnic', 282),
('Men\xa0\xa0(Multicolor)', 278),
('Multicolor', 241),
('Wear', 241),
('Formal', 238),
('Tan', 235),
('Synthetic', 234),
('Gold', 234),
('Canvas', 232),
('Party', 225),
('Men\xa0\xa0(Grey)', 221),
('Perfect', 220),
('Training', 218),
('IDP', 217),
('Gym', 212),
('Partywear', 206),
('Jutti', 205),
('Women\xa0\xa0(Brown)', 197),
('Grey)', 194),
('Ankle', 189),
('Shoe', 183),
('New', 182),
('Women\xa0\xa0(Red)', 180),
('Fashionable', 178),
('Black)', 175),
('Womens', 164),
('Navy', 158),
('Design', 153),
('Lightweight', 151),
('Derby', 147),
('Black,', 144),
('Women\xa0\xa0(Tan)', 144),
('White)', 142),
('Men\xa0\xa0(Red)', 141)]
```

```
train_df['combo'] =
train_df['title'].str.upper().str.contains('COMBO').astype(int)

sp_cols = ['Running','Black' , 'Boots', 'Casual', 'Bellies' ,
'Jutis' , 'Flats', 'Heels', 'Wedges', 'Stylish', 'Walking', 'Slip',
'Sports', 'Girls',
           'Leather', 'Mojaris', 'Pink', 'Lace', 'Brown', 'Grey',
'Blue', 'Pack' , 'Loafers', 'Beige', 'White', 'Red', 'Ethnic',
'Multicolor', 'Formal',
           'Tan', 'Synthetic', 'Gold', 'Canvas', 'Party',
'Training','IDP', 'Gym', 'Jutti', 'Ankle', 'New', 'Navy',
'Lightweight', 'Derby']

train_df['sandal'] =
train_df['title'].str.upper().str.contains('SANDAL').astype(int)
train_df['sneaker'] =
train_df['title'].str.upper().str.contains('SNEAKER').astype(int)
train_df['shoe'] =
train_df['title'].str.upper().str.contains('SHOE').astype(int)
train_df['chappal'] =
(train_df['title'].str.upper().str.contains('SLIPPERS') |
train_df['title'].str.upper().str.contains('SLIDES')
                    | train_df['title'].str.upper().str.contains('FLIP
FLOPS')).astype(int)

train_df.head()
```

```
      id                                              title  Rating  
platform  \
0  16695   Fashionable & Comfortable Bellies For Women   (...     3.9
0
1   5120  Combo Pack of 4 Casual Shoes Sneakers For Men   ...     3.8
0
2  18391          Cilia Mode Leo Sneakers For Women  (White)     4.4
0
3    495                            Men Black Sports Sandal     4.2
0
4  16408                            Men Green Sports Sandal     3.9
0

   price1  actprice1  Offer %  fulfilled1  total_cus  maincateg_Men  \
0     698        999     30.1           0       76.0              0
1     999       1999     50.0           1     1062.0              1
2    2749       4999     45.0           1       34.0              0
3     518        724     15.8           1    70522.0              1
4    1379       2299     40.0           1      154.0              1

   maincateg_Unknown  maincateg_Women  combo  sandal  sneaker  shoe  
chappal
0                  0                1      0       0        0     0
0
1                  0                0      1       0        1     1
0
2                  0                1      0       0        1     0
0
3                  0                0      0       1        0     0
0
4                  0                0      0       1        0     0
0
```

```python
for cl in sp_cols:
    train_df[cl] = 
train_df['title'].str.upper().str.contains(cl.upper()).astype(int)

train_df.drop('title', axis='columns', inplace=True)

train_df = train_df[(['Rating','actprice1', 'price1', 'fulfilled1', 
'sandal', 'sneaker', 'shoe', 'chappal', 
'maincateg_Men','maincateg_Women', 'platform', 
                'combo']) + sp_cols]

train_df.head()
```

```
   Rating  actprice1  price1  fulfilled1  sandal  sneaker shoe  
chappal  \
0     3.9        999     698           0       0        0    0
0
```

```
1      3.8        1999      999             1       0           1     1
0
2      4.4        4999     2749             1       0           1     0
0
3      4.2         724      518             1       1           0     0
0
4      3.9        2299     1379             1       1           0     0
0

   maincateg_Men  maincateg_Women ...    Party  Training   IDP  Gym
Jutti  \
0              0                1 ...        0         0     0    0
0
1              1                0 ...        0         0     0    0
0
2              0                1 ...        0         0     0    0
0
3              1                0 ...        0         0     0    0
0
4              1                0 ...        0         0     0    0
0

   Ankle   New  Navy   Lightweight Derby
0      0     0     0             0     0
1      0     0     0             0     0
2      0     0     0             0     0
3      0     0     0             0     0
4      0     0     0             0     0

[5 rows x 55 columns]
```

```
x = train_df.drop(['price1'],axis=1)
y = train_df['price1']
```

```
x.shape,y.shape
```

```
((15730, 54), (15730,))
```

Let's train the models:

```
dt_model = DecisionTreeRegressor()
score = cross_val_score(dt_model,x,y,cv=5).mean()
score
```

```
0.864176135911001
```

```
lgb_model = LGBMRegressor()
score_lgb = cross_val_score(lgb_model,x,y).mean()
score_lgb
```

```
0.8798050128232864
```

```
xgb_model = XGBRegressor()
score_xgb = cross_val_score(xgb_model,x,y).mean()
score_xgb

0.8986259678187778
```