

A modern bedroom interior featuring a low-profile wooden bed with a grey upholstered headboard. The bed is dressed in a light grey duvet and two black pillows. On either side of the bed are matching wooden bedside tables. To the left of the bed, a large potted plant with long green leaves sits on a white surface. A framed abstract painting is leaning against the wall on the far left. Two pendant lights with white shades and gold-colored stems hang above the right bedside table. The background wall is a dark, textured grey. The floor is made of light-colored wood planks.

E-COMMERCE FURNITURE SALES DATA ANALYSIS

Insights from the 2024 Furniture Dataset

Sankalp Pandey
UMDS0534

Project Overview



- **Objective:** Predict the number of furniture items sold (sold) based on product attributes such as productTitle, originalPrice, price, and tagText.
- **Tech Stack:** Python, pandas, scikit-learn, matplotlib, seaborn



Importing Libraries & Modules

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```



Load The Dataset

```
data=pd.read_csv('ecommerce_furniture_dataset_2024.csv')  
data.shape
```

```
(2000, 5)
```



Data Inspection

```
data.head()
```

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	NaN	\$46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	NaN	\$169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	\$78.4	\$39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	NaN	\$111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	\$48.82	\$21.37	1	Free shipping





Data Inspection

`data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   productTitle    2000 non-null   object 
 1   originalPrice   487 non-null    object 
 2   price           2000 non-null   object 
 3   sold            2000 non-null   int64  
 4   tagText         1997 non-null   object 
dtypes: int64(1), object(4)
memory usage: 78.3+ KB
```

Data Preprocessing

```
# Finding Duplicates  
data.duplicated().sum()
```

```
np.int64(0)
```

```
#Dropping the duplicated values  
data.drop_duplicates(inplace=True)
```

```
data.fillna(0)
```

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	0	\$46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	0	\$169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	\$78.4	\$39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	0	\$111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	\$48.82	\$21.37	1	Free shipping
...
1992	Oversized Outdoor Swivel Rocker Chairs Set wit...	0	\$555.46	0	Free shipping
1993	6pcs Patio Furniture Set PE Rattan Wicker Sect...	0	\$325.83	4	Free shipping
1994	Garden Furniture 4507 (Dark Grey) Steel Frame ...	0	\$105.16	2	Free shipping
1998	Furniture Acrylic Coffee Table Transparent Liv...	0	\$228.18	0	Free shipping
1999	Bed Frane Bamboo and Metal Platform Bed Frame ...	0	\$99.48	0	Free shipping

1906 rows × 5 columns

Data Preprocessing

```
#Describing the data for better understanding
data.describe()
```

	sold
count	1906.000000
mean	24.449633
std	260.247848
min	0.000000
25%	1.000000
50%	3.000000
75%	9.750000
max	10000.000000

```
# Remove '$' and ',' symbols, then convert to float
data['originalPrice'] = data['originalPrice'].astype(str).str.replace('$', '', regex=True).replace('nan', None).astype(float)
data['price'] = data['price'].astype(str).str.replace('$', '', regex=True).astype(float)
```

```
# Display the updated data types and first few rows
data.dtypes
```

```
productTitle    object
originalPrice    float64
price            float64
sold             int64
tagText          object
dtype: object
```

```
# Filling Original Price column by assuming 30% of Price column
data['originalPrice'] = data['originalPrice'].fillna(data['price'] * 1.3)
data.head(5)
```

	productTitle	originalPrice	price	sold	tagText
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	60.827	46.79	600	Free shipping
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	220.636	169.72	0	Free shipping
2	Desser For Bedroom With 7 Fabric Drawers Organ...	78.400	39.46	7	Free shipping
3	Modern Accent Boucle Chair,Upholstered Tufted ...	145.587	111.99	0	Free shipping
4	Small Unit Simple Computer Desk Household Wood...	48.820	21.37	1	Free shipping

Data Preprocessing

```
# Extract shipping cost or set to 0 for "Free shipping"
data['shippingCost'] = data['tagText'].str.extract(r'(\d+\.\d+)') # Extract numeric values
data['shippingCost'] = data['shippingCost'].fillna(0).astype(float)
data.drop# Convert to float, replace NaN with 0
# Display the updated DataFrame
data.head()
```

	productTitle	originalPrice	price	sold	tagText	shippingCost
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	60.827	46.79	600	Free shipping	0.0
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	220.636	169.72	0	Free shipping	0.0
2	Desser For Bedroom With 7 Fabric Drawers Organ...	78.400	39.46	7	Free shipping	0.0
3	Modern Accent Boucle Chair,Upholstered Tufted ...	145.587	111.99	0	Free shipping	0.0
4	Small Unit Simple Computer Desk Household Wood...	48.820	21.37	1	Free shipping	0.0



Numerical Analysis

```
# Calculating total cost (price + shipping cost)
data['Total_Cost'] = data['price'] + data['shippingCost']
data.head()
```

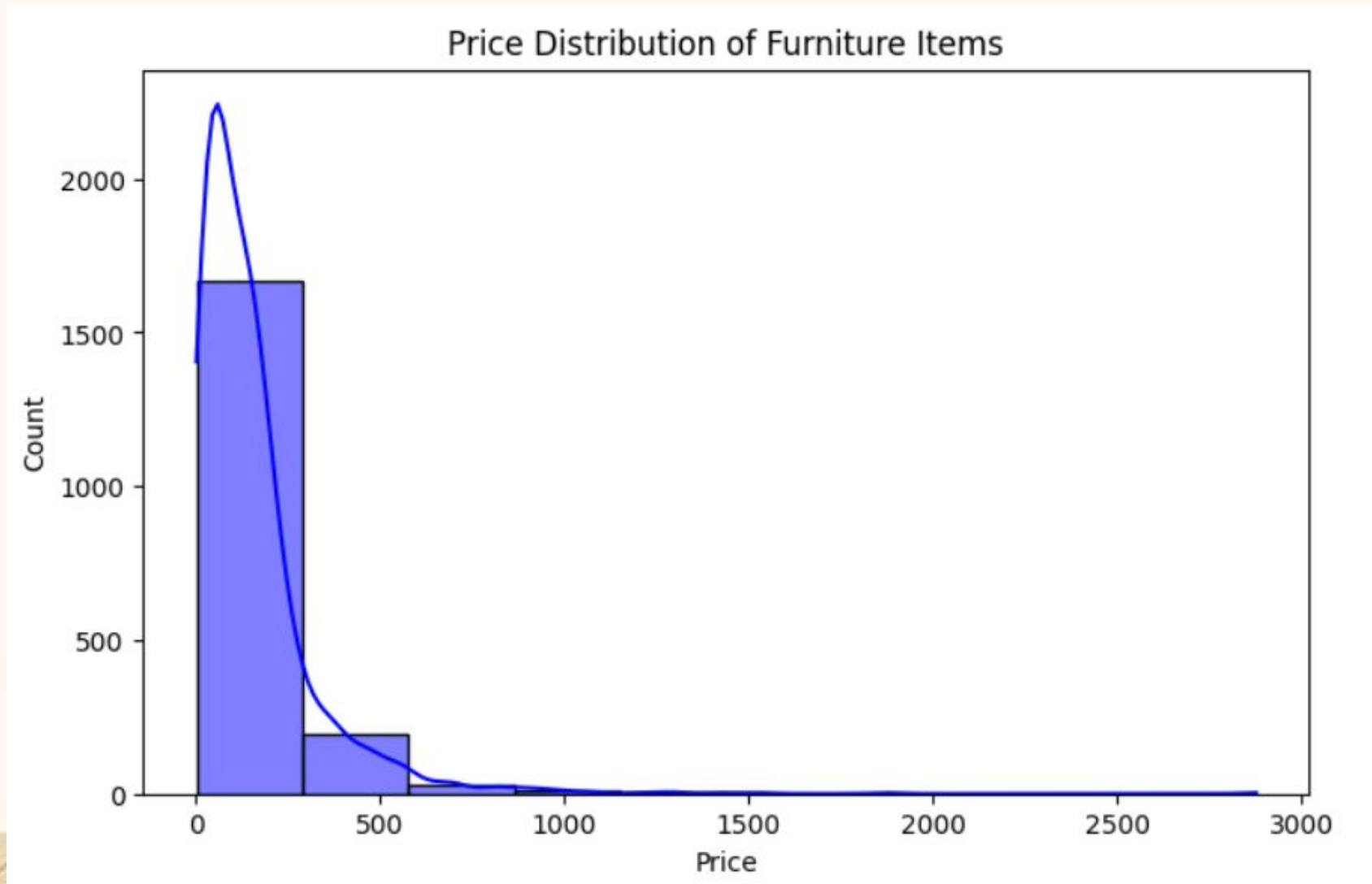
	productTitle	originalPrice	price	sold	tagText	shippingCost	Total_Cost
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	60.827	46.79	600	Free shipping	0.0	46.79
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	220.636	169.72	0	Free shipping	0.0	169.72
2	Desser For Bedroom With 7 Fabric Drawers Organ...	78.400	39.46	7	Free shipping	0.0	39.46
3	Modern Accent Boucle Chair,Upholstered Tufted ...	145.587	111.99	0	Free shipping	0.0	111.99
4	Small Unit Simple Computer Desk Household Wood...	48.820	21.37	1	Free shipping	0.0	21.37

Total sales for summing
total revenue is
\$2110806.88

```
# Calculating total revenue (price * sold)
data['Total_Revenue'] = data['price'] * data['sold']
data.head()
```

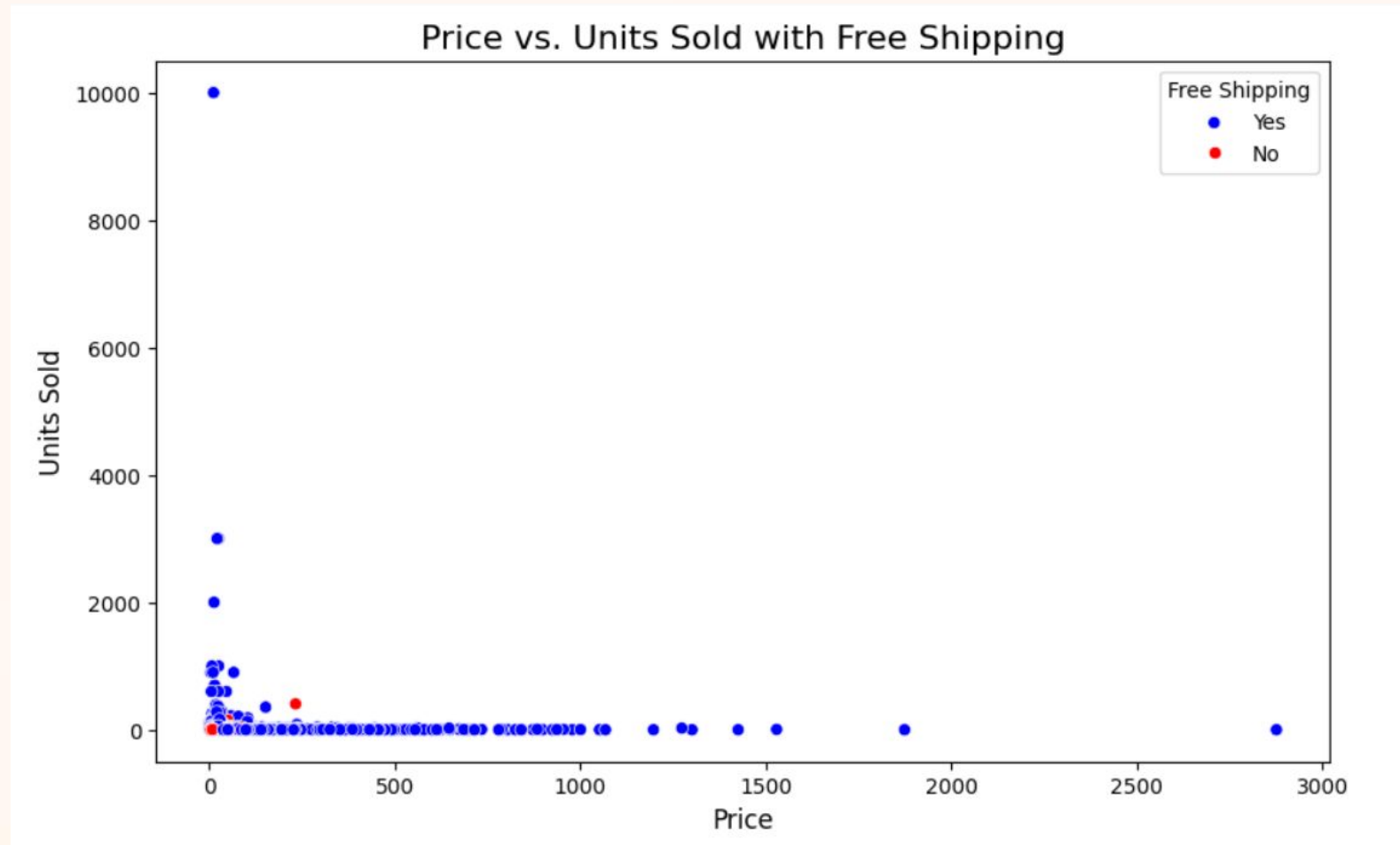
	productTitle	originalPrice	price	sold	tagText	shippingCost	Total_Cost	Total_Revenue
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	60.827	46.79	600	Free shipping	0.0	46.79	28074.00
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	220.636	169.72	0	Free shipping	0.0	169.72	0.00
2	Desser For Bedroom With 7 Fabric Drawers Organ...	78.400	39.46	7	Free shipping	0.0	39.46	276.22
3	Modern Accent Boucle Chair,Upholstered Tufted ...	145.587	111.99	0	Free shipping	0.0	111.99	0.00
4	Small Unit Simple Computer Desk Household Wood...	48.820	21.37	1	Free shipping	0.0	21.37	21.37

Exploratory Data Analysis (EDA)



The graph shows the distribution of furniture item prices. The histogram is skewed to the right, indicating that most furniture items are priced lower, with a few items at higher price points.

Exploratory Data Analysis (EDA)



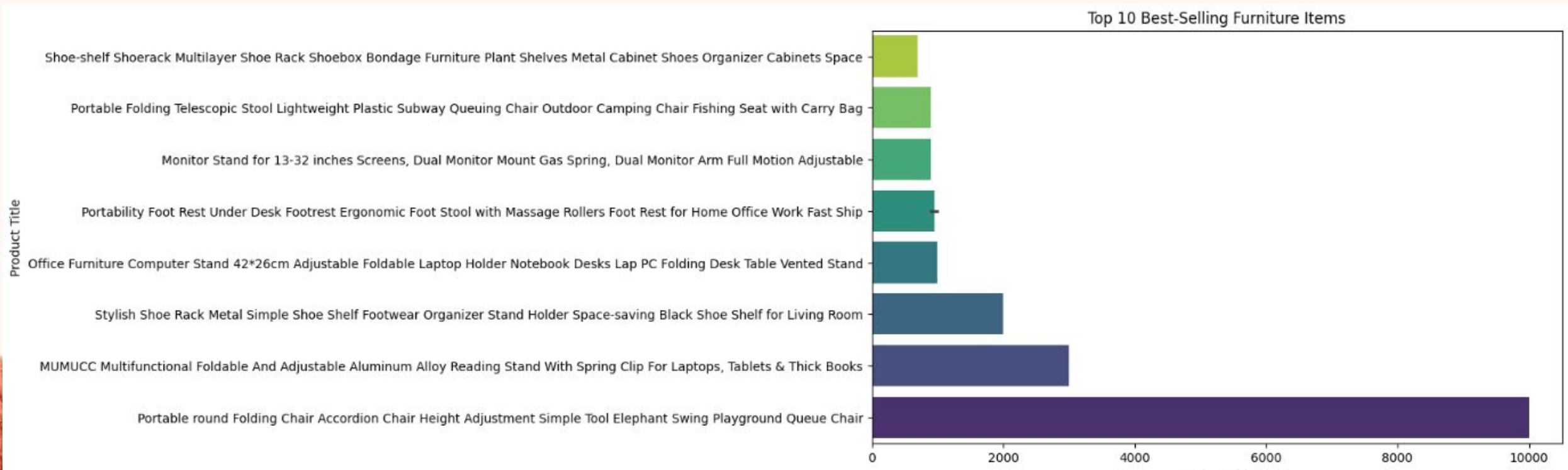
Offering that Free shipping can be a strong factor in increasing units sold, especially at lower price points. Products with free shipping tend to selling more units even at higher prices while without free shipping the number of units sold is likely to be significantly lower.

Exploratory Data Analysis (EDA)



This graph offering that the people are highly recommended free shipping even on high prices compare to paid shipping

Exploratory Data Analysis (EDA)



This graph shows the top 10 products sold and indicates the higher demanding product is portable round folding which have the maximum 10000 unit sold

Feature Engineering

```
#Calculating discountAmount as originalPrice- price and calculating discount percentage as (discountAmount/OriginalPrice)*100
```

```
data['discountAmount']=data['originalPrice']-data['price']
```

```
data['discountPercent']=(data['discountAmount']/data['originalPrice'])*100
```

```
data.head(5)
```

	productTitle	originalPrice	price	sold	tagText	shippingCost	Total_Cost	Total_Revenue	discountAmount	discountPercent
0	Dresser For Bedroom With 9 Fabric Drawers Ward...	60.827	46.79	600	Free shipping	0.0	46.79	28074.00	14.037	23.076923
1	Outdoor Conversation Set 4 Pieces Patio Furnit...	220.636	169.72	0	Free shipping	0.0	169.72	0.00	50.916	23.076923
2	Desser For Bedroom With 7 Fabric Drawers Organ...	78.400	39.46	7	Free shipping	0.0	39.46	276.22	38.940	49.668367
3	Modern Accent Boucle Chair,Upholstered Tufted ...	145.587	111.99	0	Free shipping	0.0	111.99	0.00	33.597	23.076923
4	Small Unit Simple Computer Desk Household Wood...	48.820	21.37	1	Free shipping	0.0	21.37	21.37	27.450	56.226956

Feature Engineering

```
# Convert productTitle into a numeric feature using TF-IDF Vectorizer  
from sklearn.feature_extraction.text import TfidfVectorizer  
tfidf = TfidfVectorizer(max_features=100)  
productTitle_tfidf = tfidf.fit_transform(data['productTitle'])
```

```
# Convert to DataFrame and concatenate to original df  
productTitle_tfidf_df = pd.DataFrame(productTitle_tfidf.toarray(),  
columns=tfidf.get_feature_names_out())  
df = pd.concat([data, productTitle_tfidf_df], axis=1)
```

```
# Drop original productTitle as it's now encoded  
df = df.drop('productTitle', axis=1)
```



Model Selection & Training

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Split the dataset into features (X) and target (y)
X = data[['originalPrice', 'price', 'shippingCost']]
y = data['sold']
X = X.dropna()
y = y.loc[X.index]
```

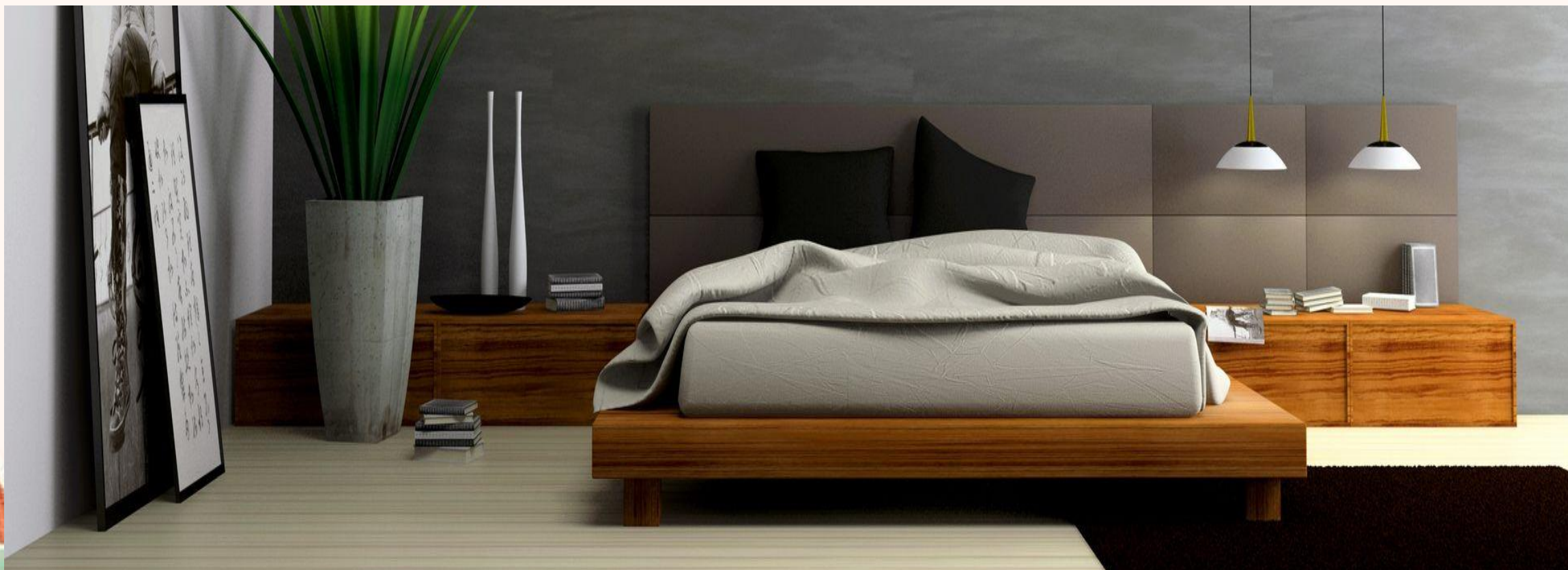
```
# Train-test split (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
lr_model = LinearRegression()
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
lr_model.fit(X_train, y_train)
rf_model.fit(X_train, y_train)
```

▼ RandomForestRegressor ⓘ ⓘ
RandomForestRegressor(random_state=42)

Model Evaluation

```
# Predict with Linear Regression
y_pred_lr = lr_model.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)
# Predict with Random Forest
y_pred_rf = rf_model.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
# Print model evaluation results
print(f'Linear Regression MSE: {mse_lr}, R2: {r2_lr}')
print(f'Random Forest MSE: {mse_rf}, R2: {r2_rf}')
```

```
Linear Regression MSE: 12842.67464534846, R2: 0.005601402889128804
Random Forest MSE: 11209.40128847979, R2: 0.13206452522299683
```

Thank You...!