TEAM GRADIENT GEEKS

Uttam Mahata, 2nd Year,CST, IIEST Shibpur

Anurag Ghosh, 2nd Year,CST, IIEST Shibpur

Siddharth Sen, 2nd Year,CST, IIEST Shibpur

Suchana Hazra, 2nd Year,CST, IIEST Shibpur

ONLINE RETAIL TRANSACTIONS ANALYSIS

Team : Gradient Geeks

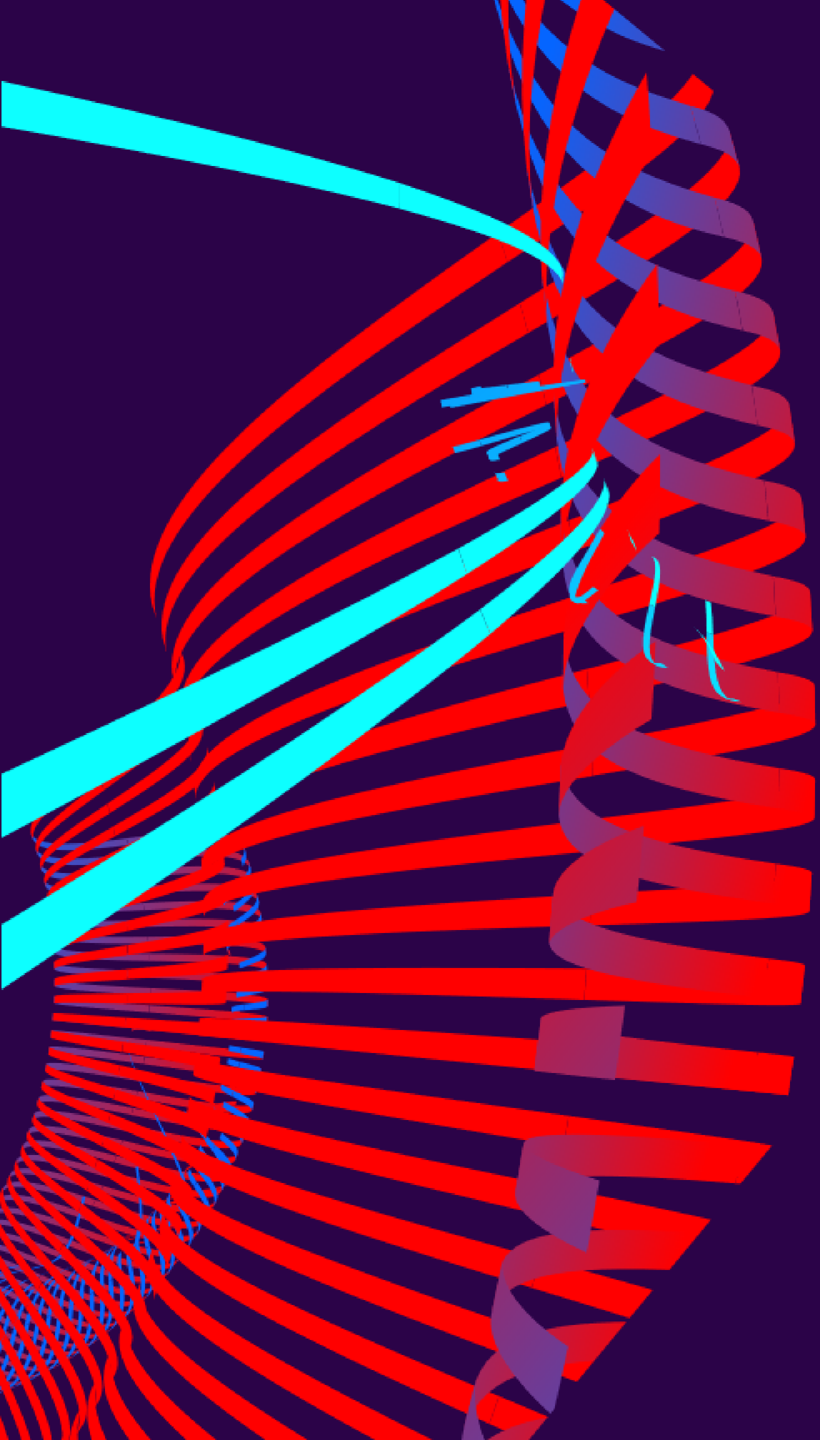
Check the codes here!

INTRODUCTION

Transaction data analysis yields insightful information that may be used to develop plans for many areas of the company. The major priority is enhancing client satisfaction is one of transaction analysis's main advantages. Businesses may find pain areas, improve the overall online buying experience, and streamline operations by carefully examining the complete consumer journey. Increased client loyalty and satisfaction are a result of this optimization. Another important factor that is influenced by transaction insights is personalization. Businesses may customize marketing campaigns by exploring the unique tastes, purchase history, and browsing behavior of each consumer. This personalization includes making product recommendations that are appropriate, creating promotions that are specifically targeted, and providing an enhanced online experience.

OBJECTIVE

In this challenge, the objective is to analyze and derive insights from two distinct datasets related to online retail transactions.



DATA PREPROCESSING AND EXPLORATION

1. The customer id was not required, so we dropped the column.
2. We checked the shape of data, i.e. the total number of rows / samples and the features available
3. Explored various statistical measures of the data
4. Checked if there exist any missing values, i.e. null values
5. Preprocessing is the key to successful model evaluation.

DATA EXPLORATION

```
data.head()
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash	Fortnightly
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card	Weekly
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal	Weekly
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal	Annually

```
[ ] data.shape
```

(3900, 18)

```
[ ] data.describe()
```

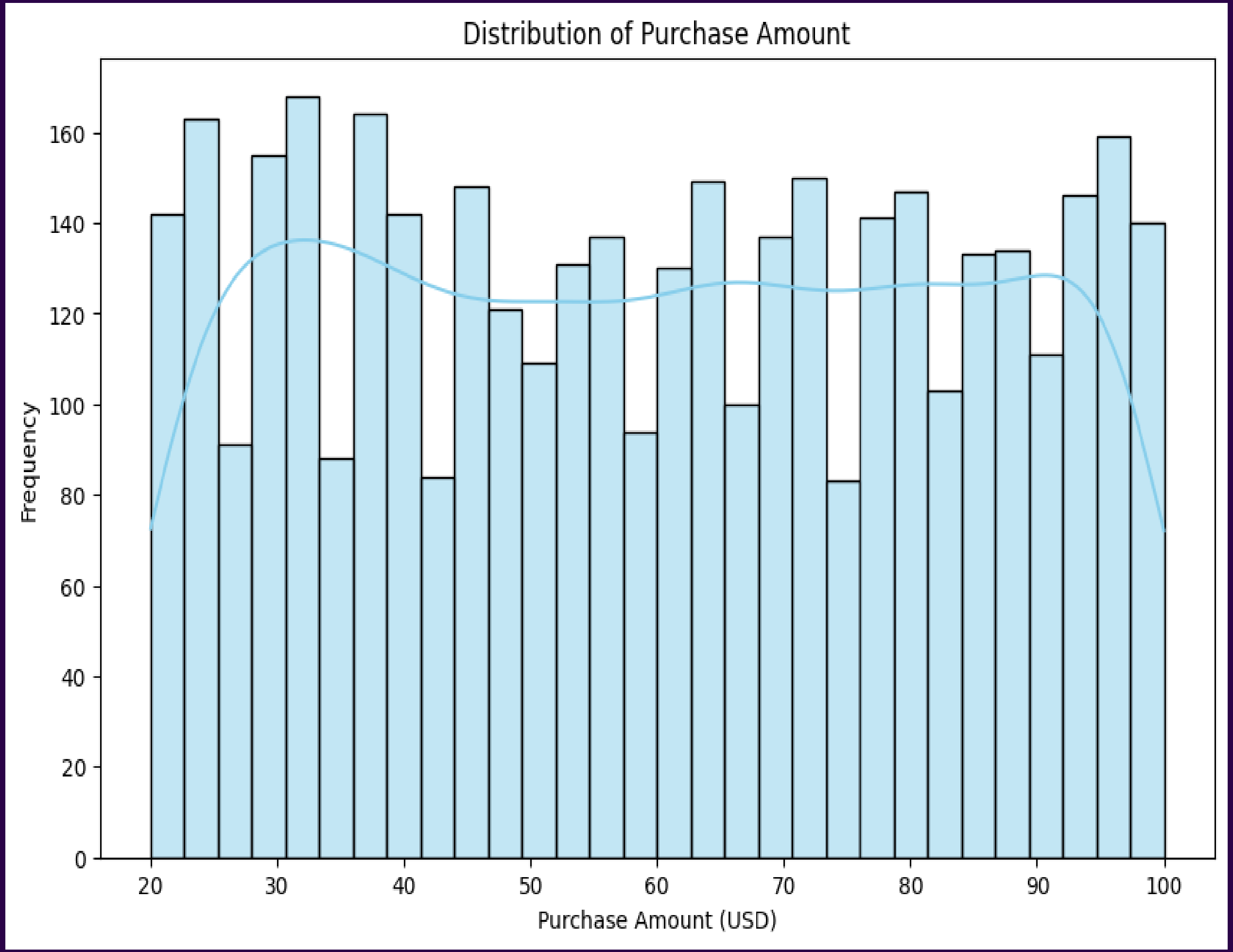
	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3900.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.749949	25.351538
std	1125.977353	15.207589	23.685392	0.716223	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.700000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

```
data.isnull().sum()

Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

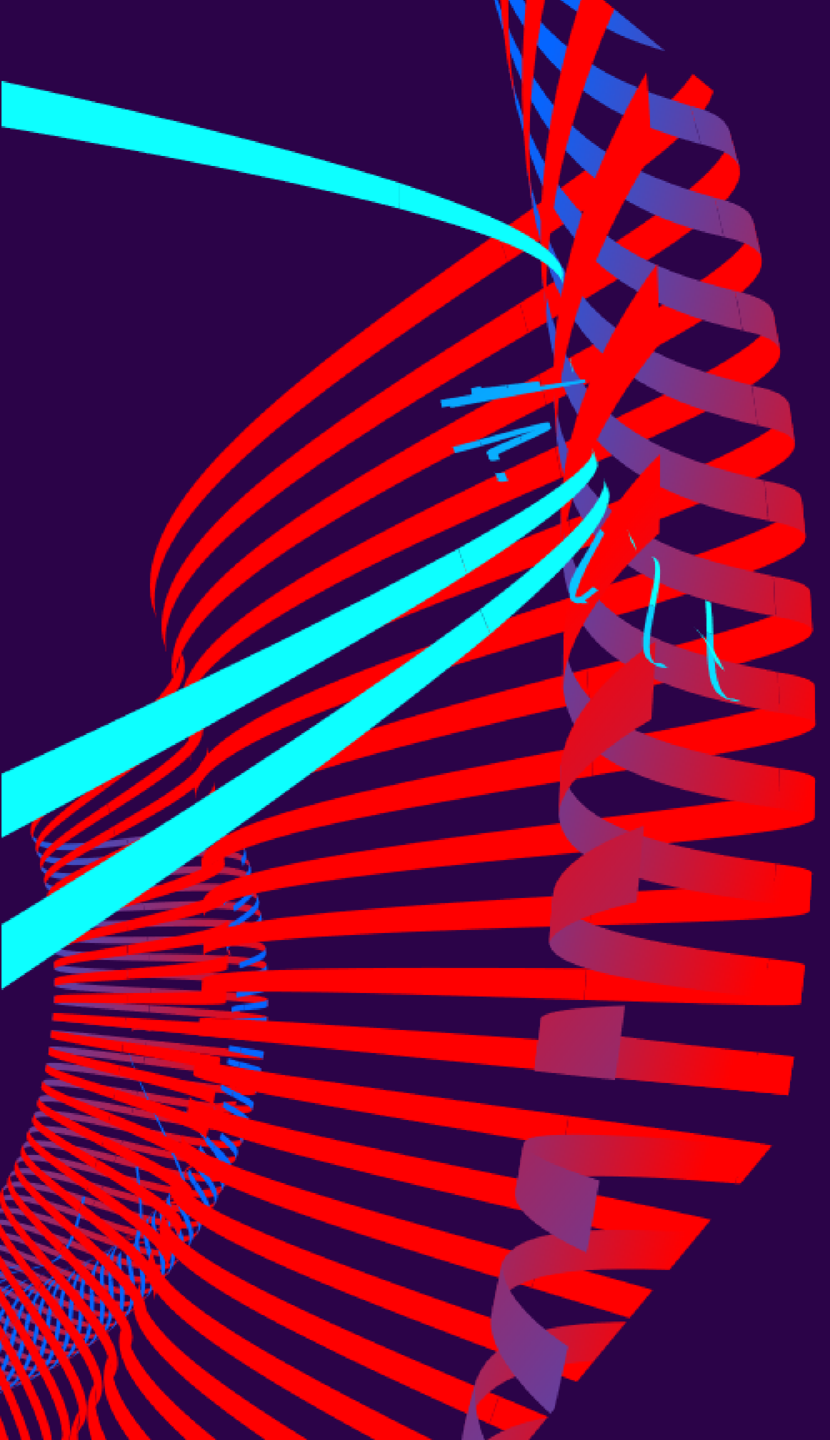
```
data.nunique()

Customer ID               3900
Age                         53
Gender                       2
Item Purchased              25
Category                     4
Purchase Amount (USD)       81
Location                    50
Size                         4
Color                       25
Season                       4
Review Rating               26
Subscription Status          2
Shipping Type                6
Discount Applied             2
Promo Code Used              2
Previous Purchases          50
Payment Method               6
Frequency of Purchases       7
dtype: int64
```

NO DUPLICATE VALUES ENCOUNTERED

DATA EXPLORATION

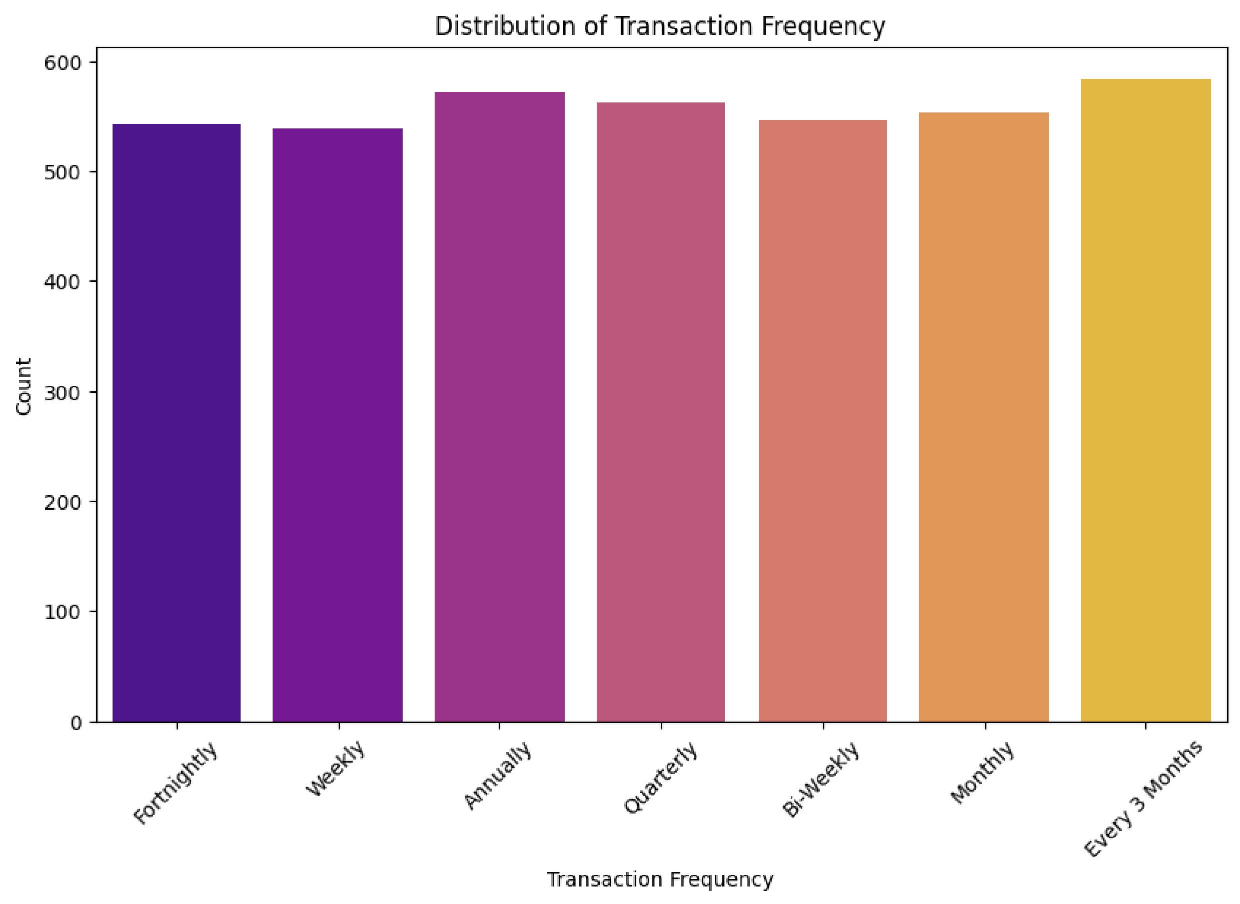


DATA EXPLORATION

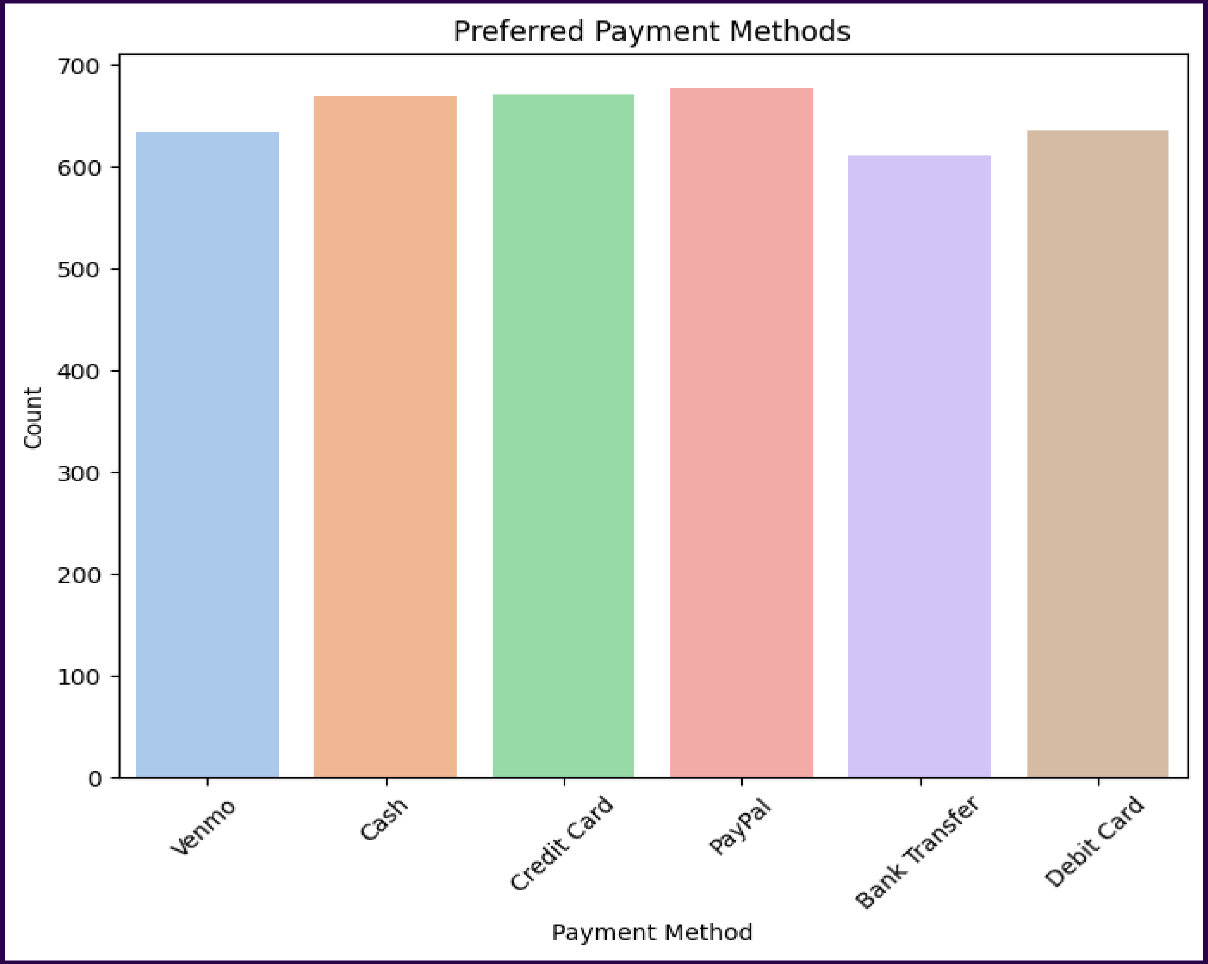


DATA EXPLORATION

BASED ON TRANSACTION FREQUENCY



DATA EXPLORATION



DATA ANALYSIS

1. TOP-SELLING PRODUCTS AND CATEGORIES

Identify and analyse the top-selling products and categories based on transaction frequency or revenue.

TWO METHODS APPLIED:

BASED ON COUNT OF OBJECTS

```
data.keys()
item_counts = data['Item Purchased'].value_counts()
print(item_counts)
```

```
Blouse         171
Jewelry        171
Pants          171
Shirt          169
Dress          166
Sweater        164
Jacket         163
Belt           161
Sunglasses     161
Coat           161
Sandals        160
Socks          159
Skirt          158
Shorts         157
Scarf          157
Hat            154
Handbag        153
Hoodie         151
Shoes          150
T-shirt        147
Sneakers       145
Boots          144
Backpack       143
Gloves         140
Jeans          124
Name: Item Purchased, dtype: int64
```

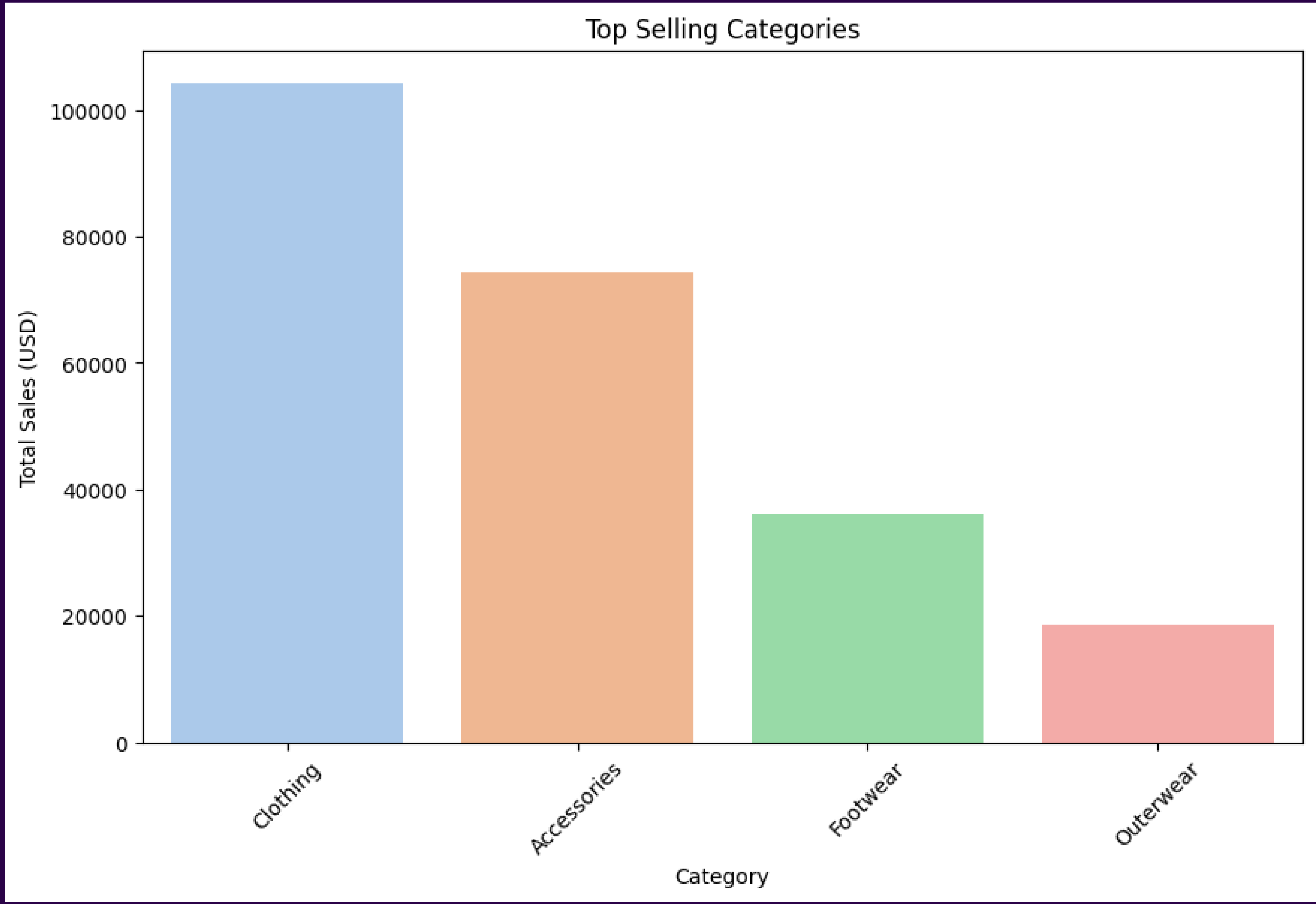
<<<< TOP PRODUCTS

BASED ON TOTAL AMOUNT OF EACH ITEM PURCHASED

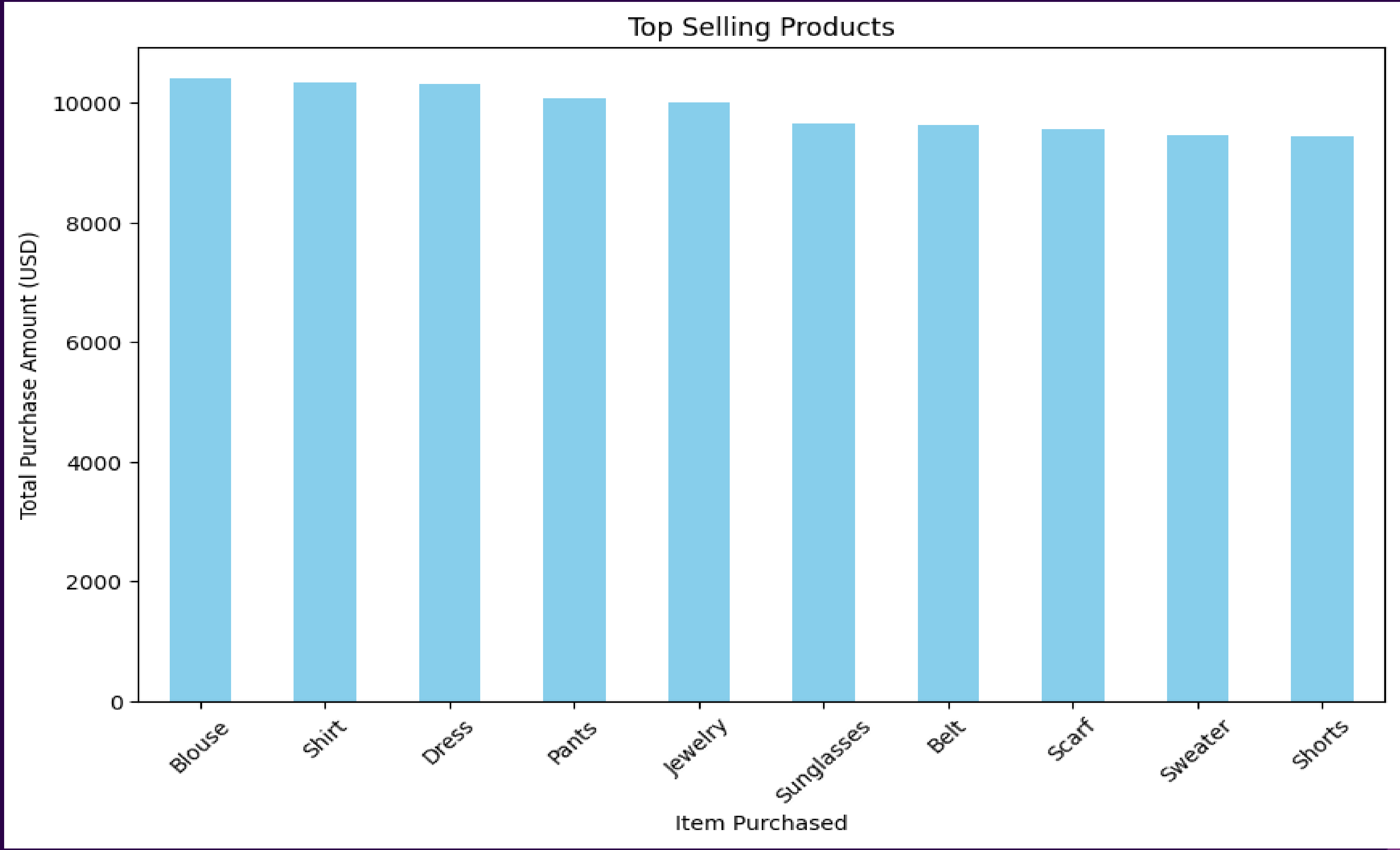
```
df = pd.DataFrame(data)
grouped_by_item = df.groupby('Item Purchased')
total_purchase_per_item = grouped_by_item['Purchase Amount (USD)'].sum()
print(total_purchase_per_item)
```

```
Item Purchased
Backpack       8636
Belt           9635
Blouse        10410
Boots          9018
Coat           9275
Dress         10320
Gloves         8477
Handbag        8857
Hat            9375
Hoodie         8767
Jacket         9249
Jeans          7548
Jewelry       10010
Pants         10090
Sandals        9200
Scarf          9561
Shirt         10332
Shoes          9240
Shorts         9433
Skirt          9402
Sneakers       8635
Socks          9252
Sunglasses     9649
Sweater        9462
T-shirt        9248
Name: Purchase Amount (USD), dtype: int64
```

TOP SELLING CATEGORY:CLOTHING



TOTAL AMOUNT PURCHASED PER CATEGORY

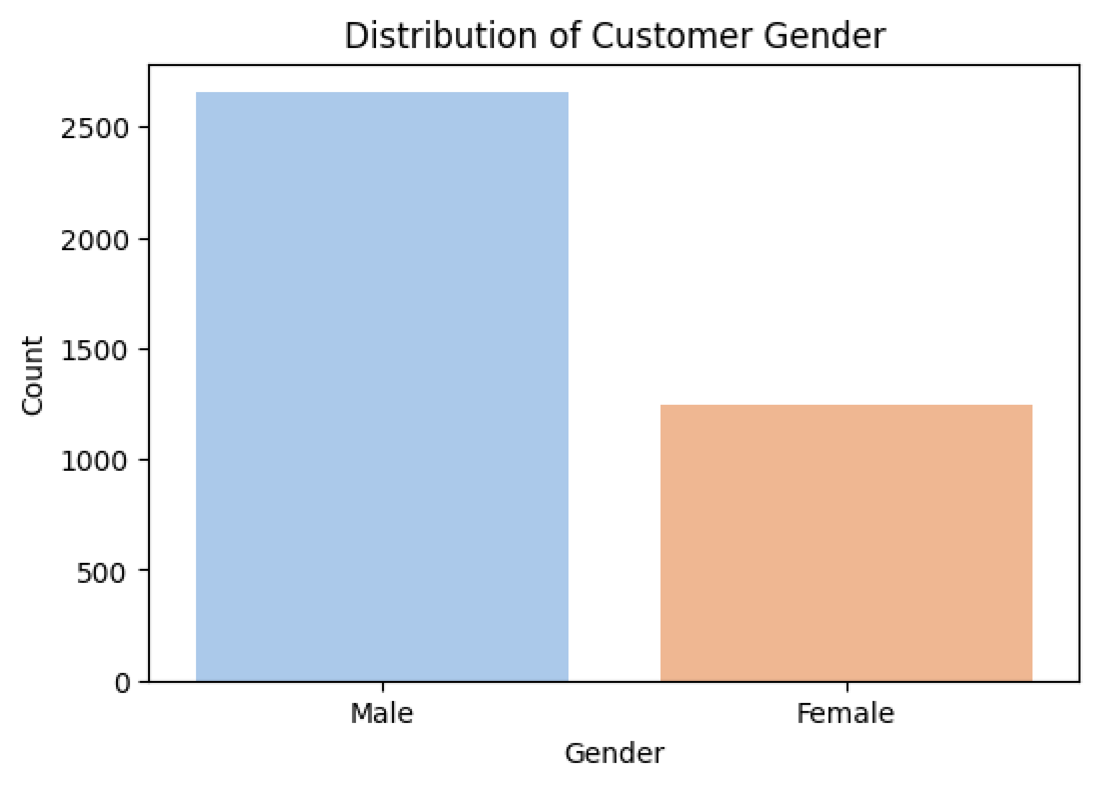


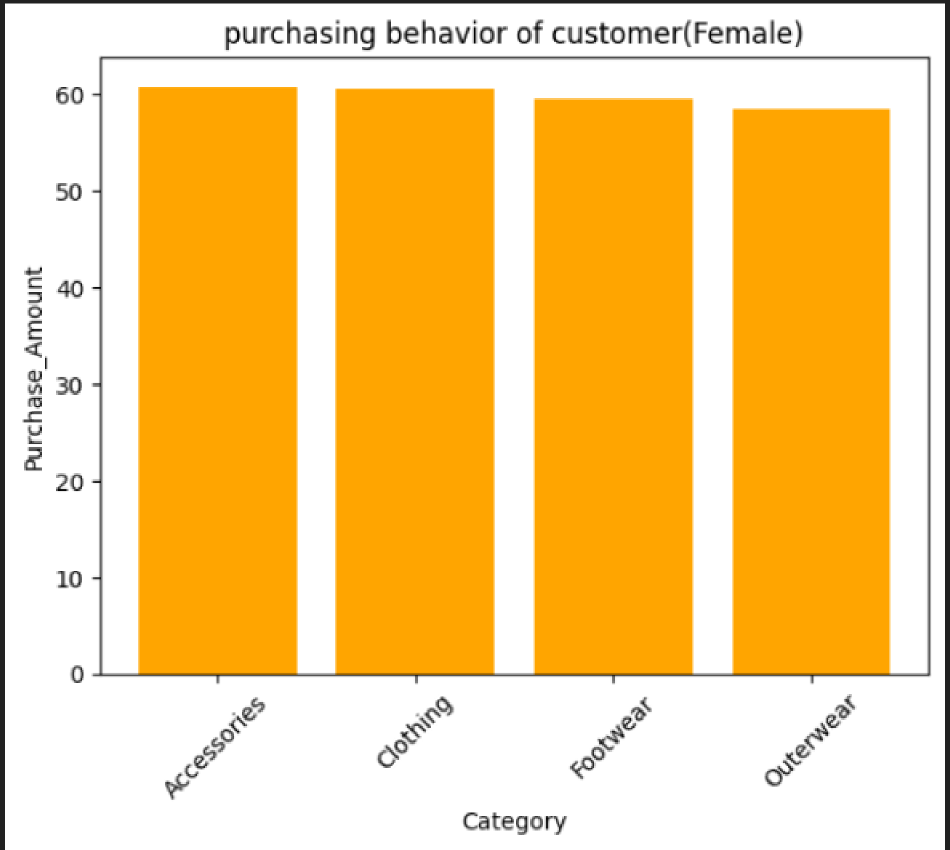
2. CUSTOMER PURCHASING BEHAVIOR PATTERNS

Investigate trends in customer purchasing behavior based on demographics (age, gender, etc.).

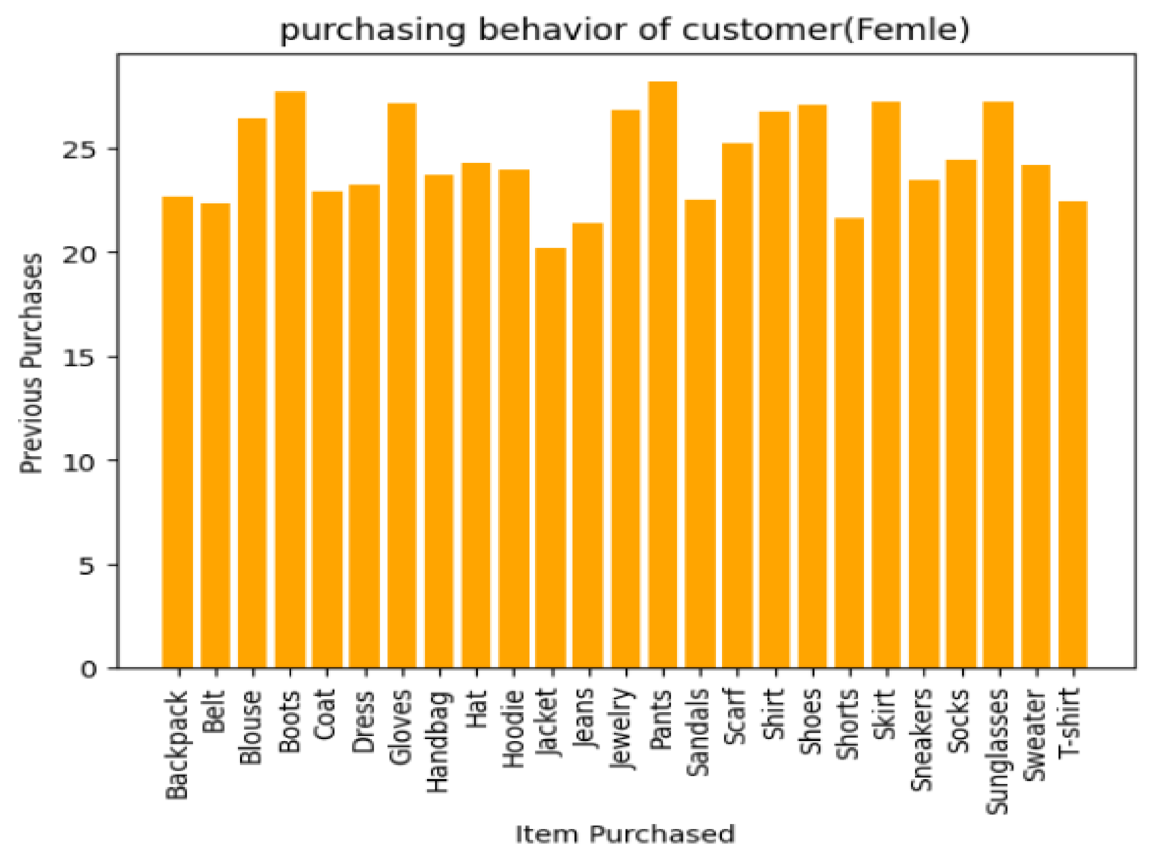
ANALYSIS OF CUSTOMERS TRENDS BASED ON GENDER

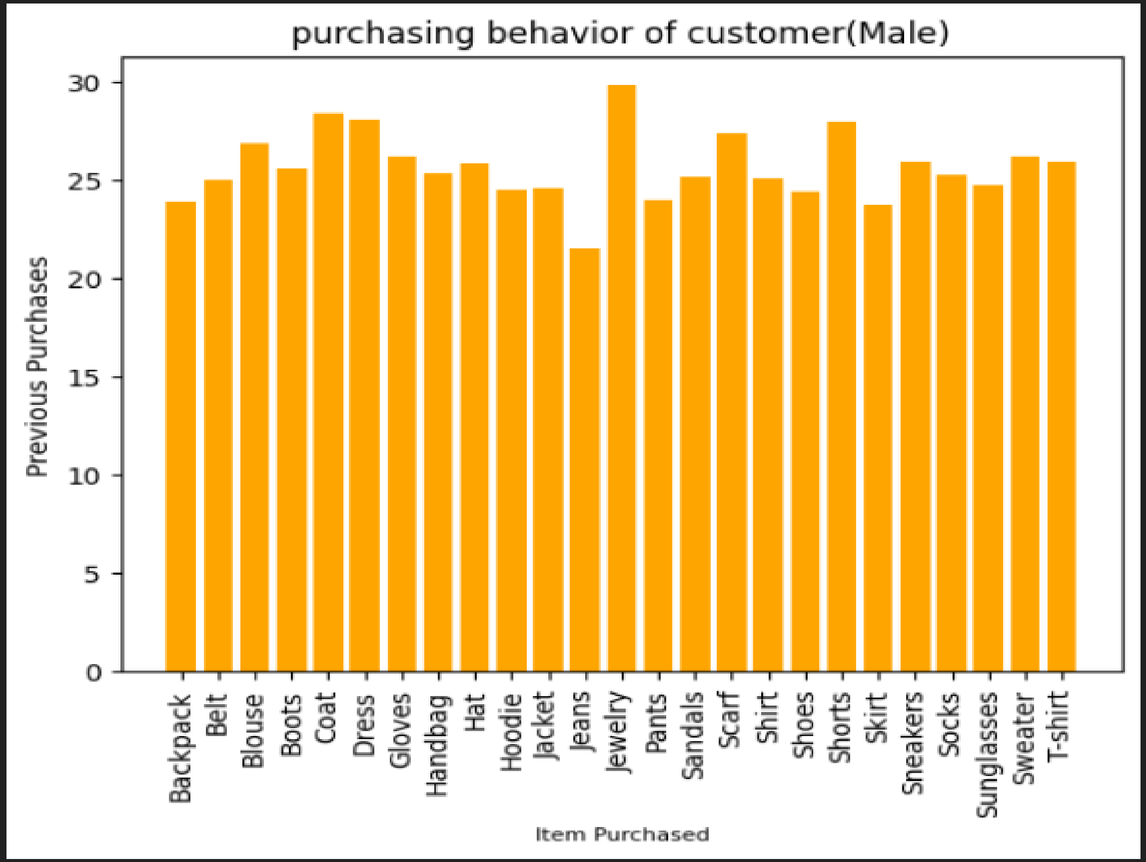
```
plt.figure(figsize=(6, 4))
sns.countplot(x='Gender', data=data, palette='pastel')
plt.title('Distribution of Customer Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```





CATEGORY WISE COMPARISON OF PURCHASING BEHAVIOUR





Comparison of the purchasing behaviours of Male and Female Customers

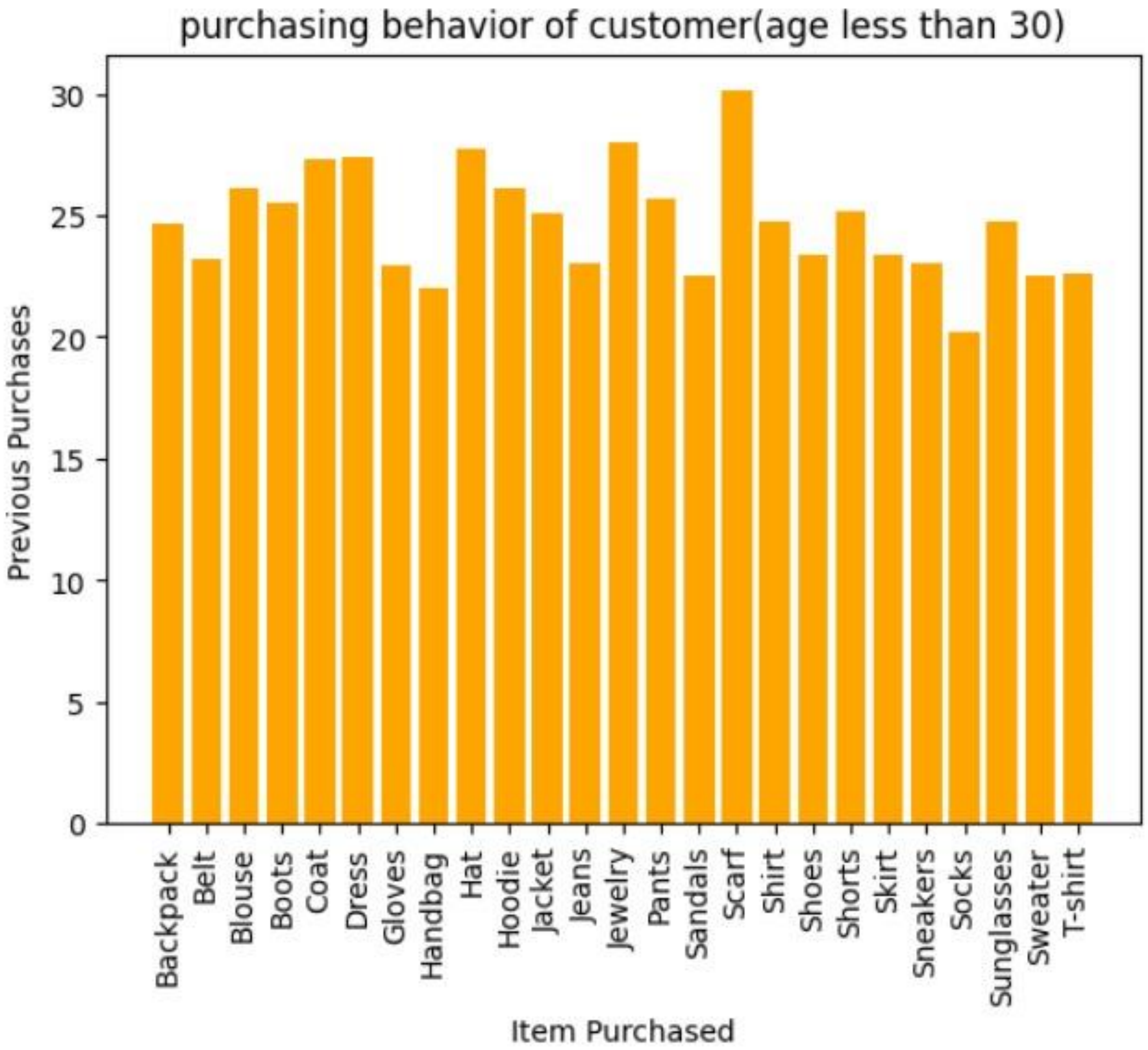
```python
avg_purchase_amount_female = dfFemale.groupby('Category')['Purchase Amount (USD)'].mean().reset_index()
plt.bar(avg_purchase_amount_female['Category'], avg_purchase_amount_female['Purchase Amount (USD)'], color = 'orange')
plt.title('purchasing behavior of customer(Female)')
plt.xlabel('Category')
plt.ylabel('Purchase_Amount')
plt.xticks(rotation=45)
plt.show()
```

```python
avg_purchase_amount_male = dfMale.groupby('Category')['Purchase Amount (USD)'].mean().reset_index()
plt.bar(avg_purchase_amount_male['Category'], avg_purchase_amount_male['Purchase Amount (USD)'], color = 'orange')
plt.title('purchasing behavior of customer(Male)')
plt.xlabel('Category')
plt.ylabel('Purchase_Amount')
plt.xticks(rotation=45)
plt.show()
```

CODE SNIPPETS : PURCHASING BEHAVIOUR OF MALE AND FEMALE CUSTOMERS

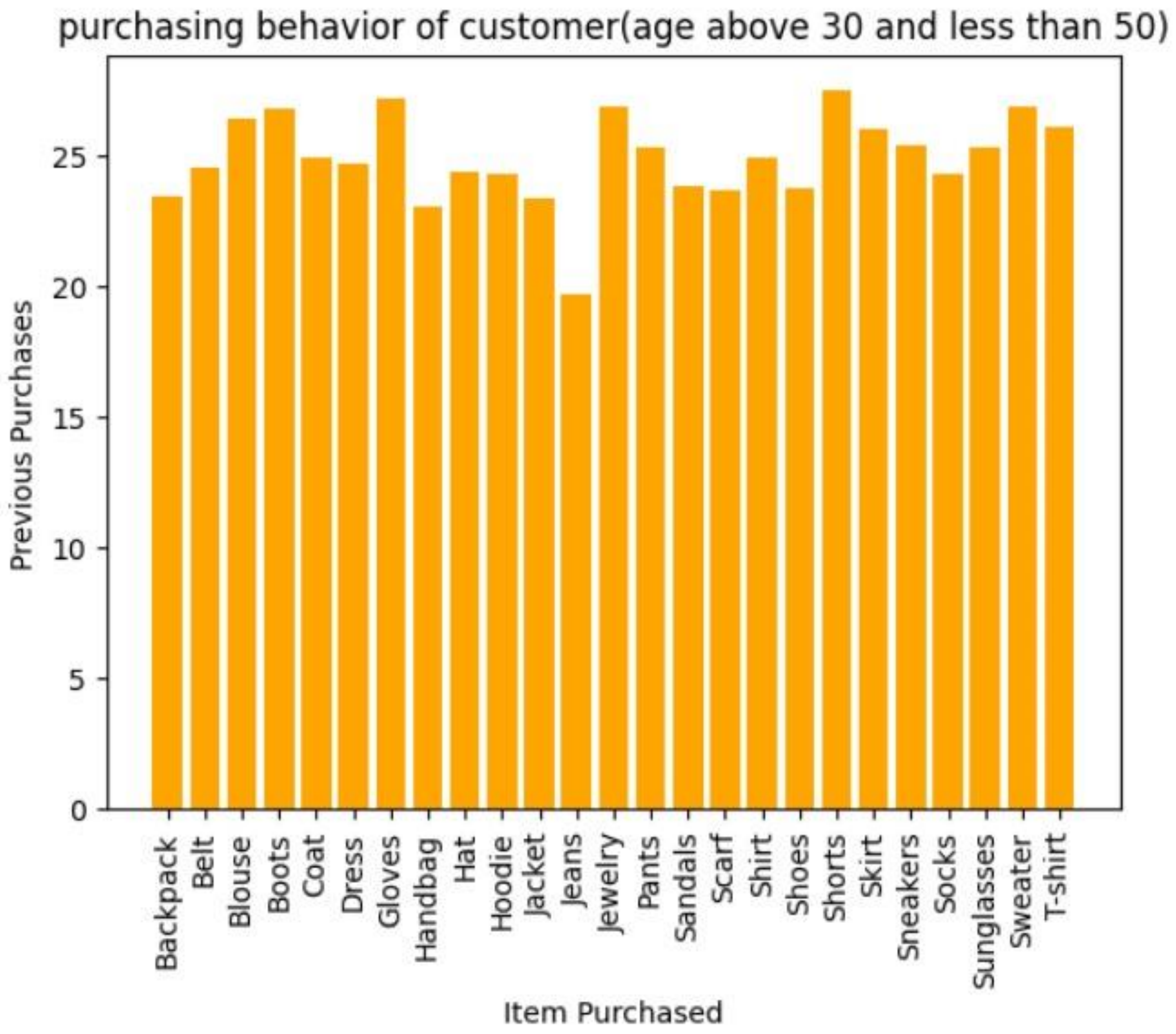
PURCHASING BEHAVIOUR VS AGE

[60]:
```python
avg_purchase_amount = dfA1.groupby('Item Purchased')['Previous Purchases'].mean().reset_index()

plt.bar(avg_purchase_amount['Item Purchased'], avg_purchase_amount['Previous Purchases'], color='orange')
plt.title('purchasing behavior of customer(age less than 30)')
plt.xlabel('Item Purchased')
plt.ylabel('Previous Purchases')
plt.xticks(rotation=90)
plt.show()
```



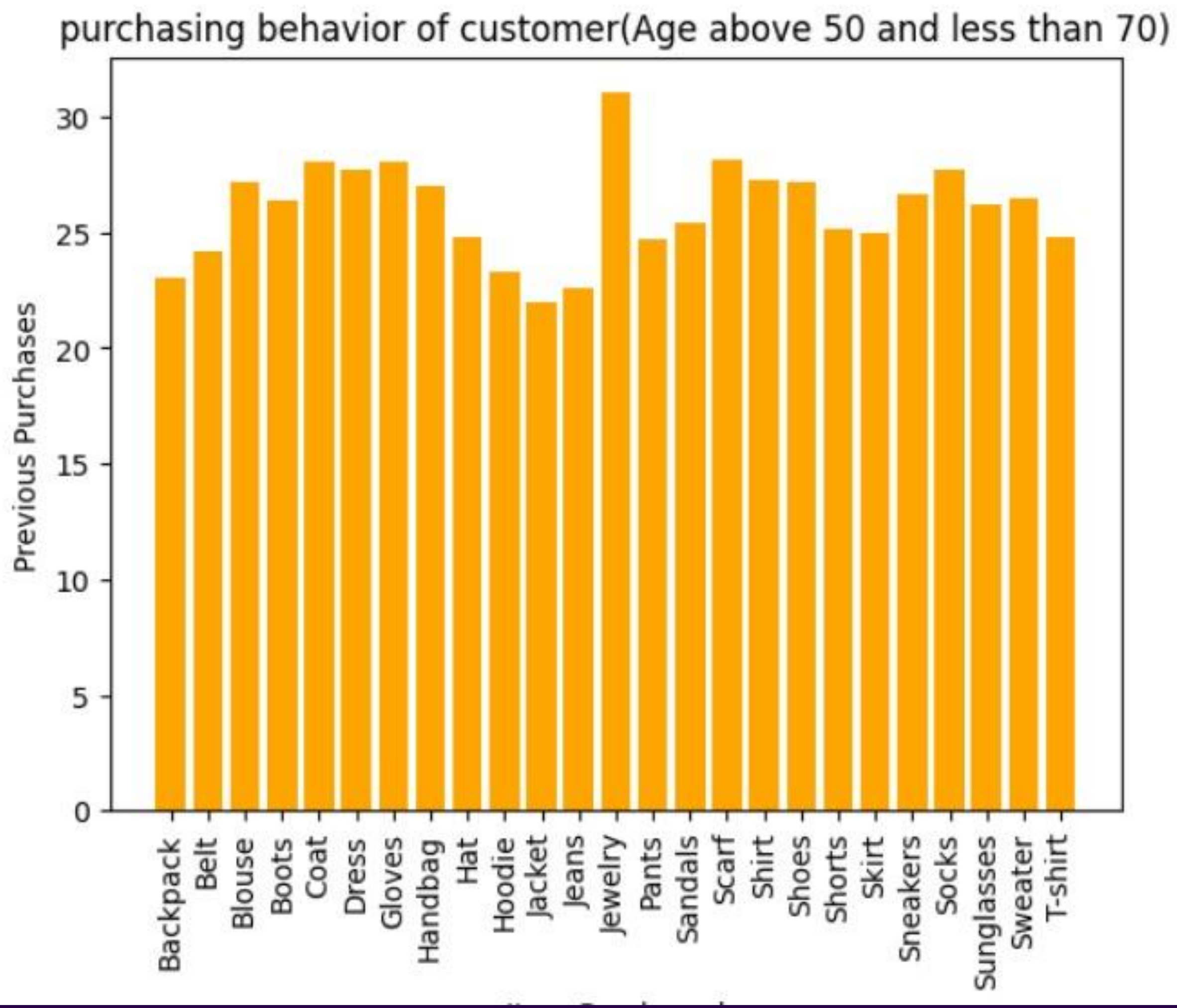
PURCHASING BEHAVIOUR VS AGE

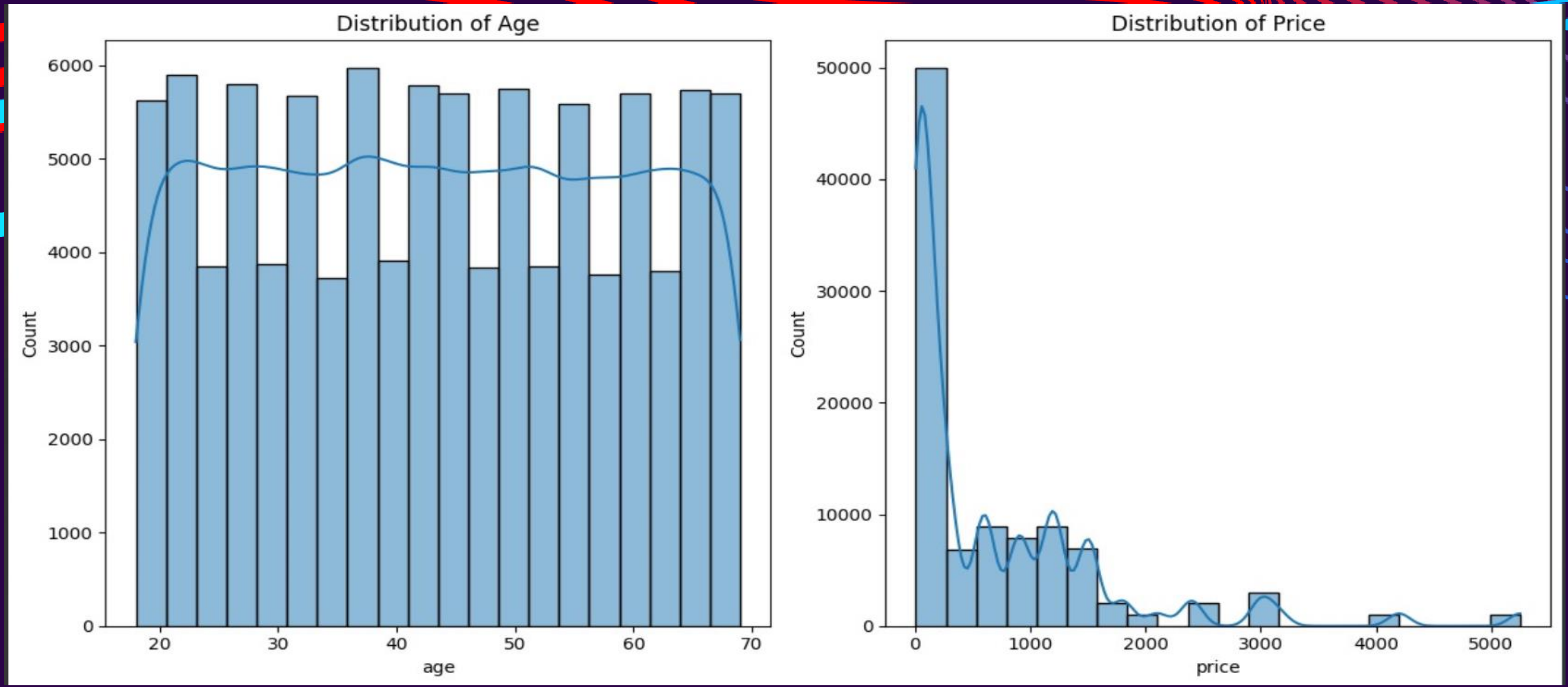
```
[62]: avg_purchase_amount = dfA2.groupby('Item Purchased')['Previous Purchases'].mean().reset_index()

plt.bar(avg_purchase_amount['Item Purchased'], avg_purchase_amount['Previous Purchases'], color='orange')
plt.title('purchasing behavior of customer(age above 30 and less than 50)')
plt.xlabel('Item Purchased')
plt.ylabel('Previous Purchases')
plt.xticks(rotation=90)
plt.show()
```



PURCHASING BEHAVIOUR VS AGE

```
[63]: dfA3 = dataset[(dataset['Age'] <= 70) & (dataset['Age'] >= 50)]

[64]: avg_purchase_amount = dfA3.groupby('Item Purchased')['Previous Purchases'].mean().reset_index()

      plt.bar(avg_purchase_amount['Item Purchased'], avg_purchase_amount['Previous Purchases'], color='orange')
      plt.title('purchasing behavior of customer(Age above 50 and less than 70)')
      plt.xlabel('Item Purchased')
      plt.ylabel('Previous Purchases')
      plt.xticks(rotation=90)
      plt.show()
```





The bar plots shown above are segregating customers on the basis of their age and the price with which they are purchasing the items.

Comparison of the purchasing behaviours of Male and Female Customers

3.CUSTOMER RATINGS AND PRODUCT SATISFACTION

ABOUT THE PROBLEM

- The features which have the potential to affect customer ratings and satisfaction are :

  Review Rating
- Purchase Amount (USD)
- Discount Applied
- Previous Purchases
- Shipping Type
- Category

We have to segment the customers based on the review ratings.

PROBLEM STATEMENT

Customer Segmentation

We need to segment the customers on the basis of their previous purchases as well as review ratings.

Customer Loyalty

We need to know the ratings given by customers who are more loyal.

Costs

These analytics would help the managers to improve the quality of the products of the supermarket.

Financials

Insights into these analytics would help in improving the business model for the supermarket.

SOLUTION FOR CUSTOMER SEGMENTATION

Find the products with the lowest review rating

The quality of these products needs to be improved

Find the products with the highest review rating

The sales of these products need to be scaled up.

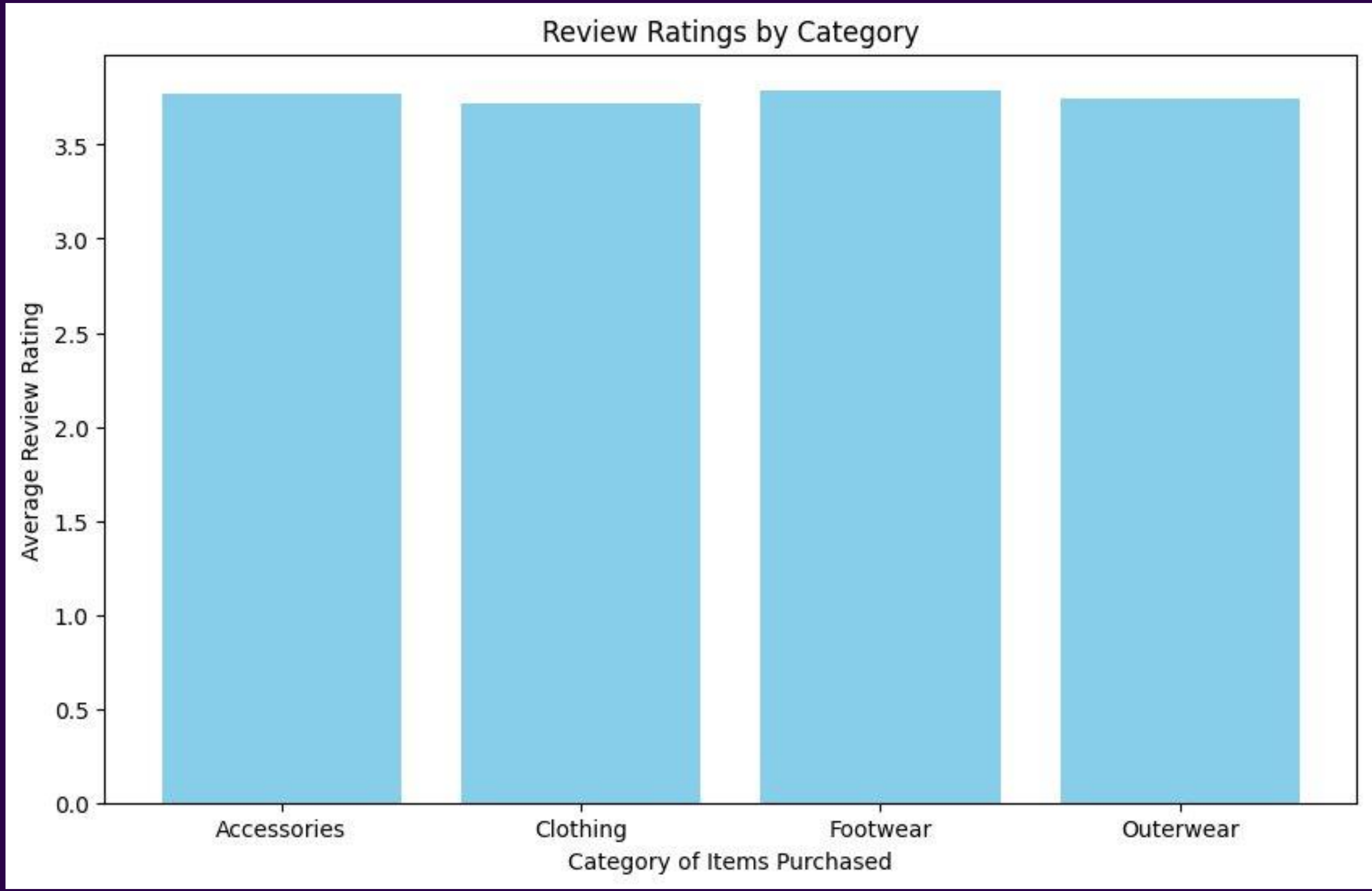
Performing K Means Clustering using the relevant features

Understand which segment of customers are giving a higher rating and which are giving a lower rating

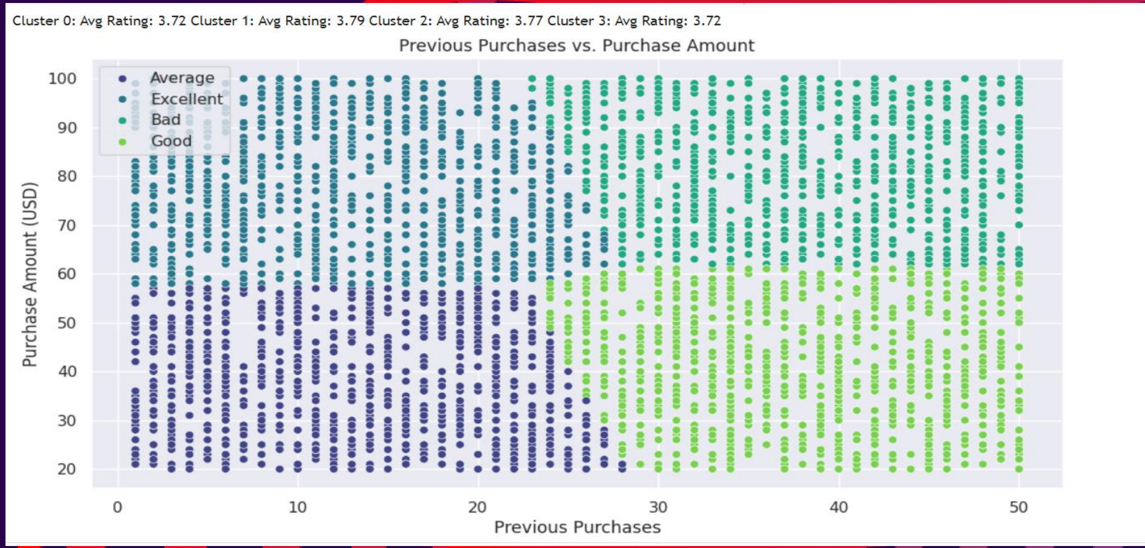
Plotting the pair plot and the scatter plots for the relevant features to see clustering using Seaborn

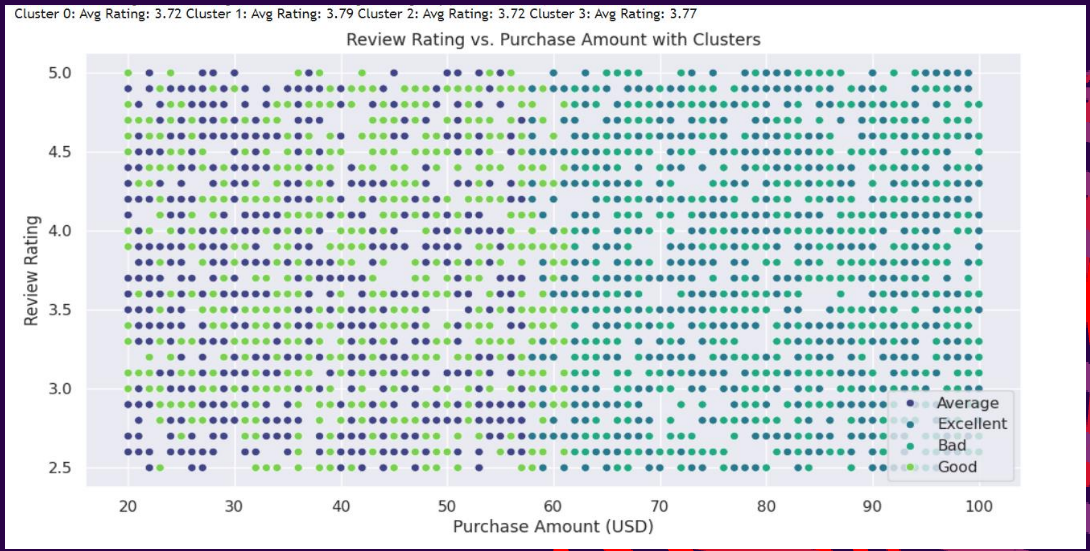
Simple analysis that gives customers the targeted information they need

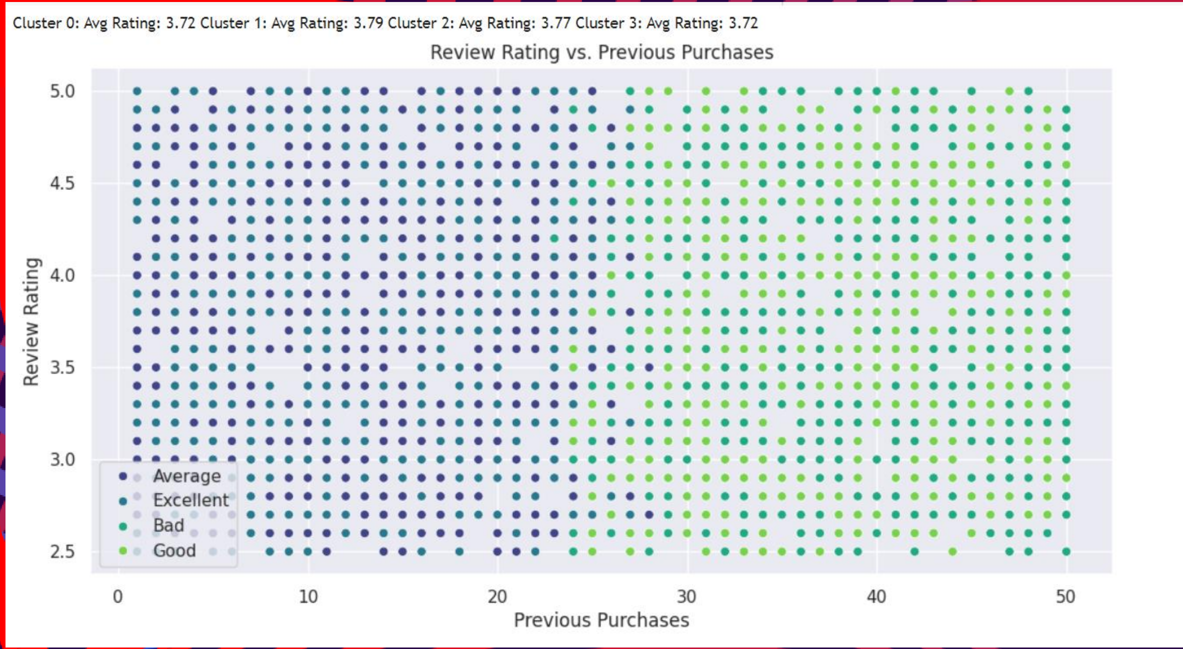
CUSTOMER RATINGS



CLUSTERS FORMED AFTER APPLYING K MEANS CLUSTERING ALGORITHM







INSIGHTS FROM THE SCATTER PLOTS

Review Rating vs Previous Purchases

- Previous purchases < 25 implies Average or Excellent Customer Rating

  Previous purchases > 25 implies Bad or Good Rating

- Customers who have been buying more over a period of time are somehow not satisfied with some of the products

Purchase Amount vs Previous Purchases

- Category 1 – Low previous purchases and Low purchase Amount customers gave average rating
- Category 2 – Low previous purchases and High Purchase Amount customers are giving excellent rating
- Category 3 – High Purchase amount and high previous purchases customers gave Bad rating.
- Category 4 – Low Purchase amount and high previous purchases customers gave Good rating.

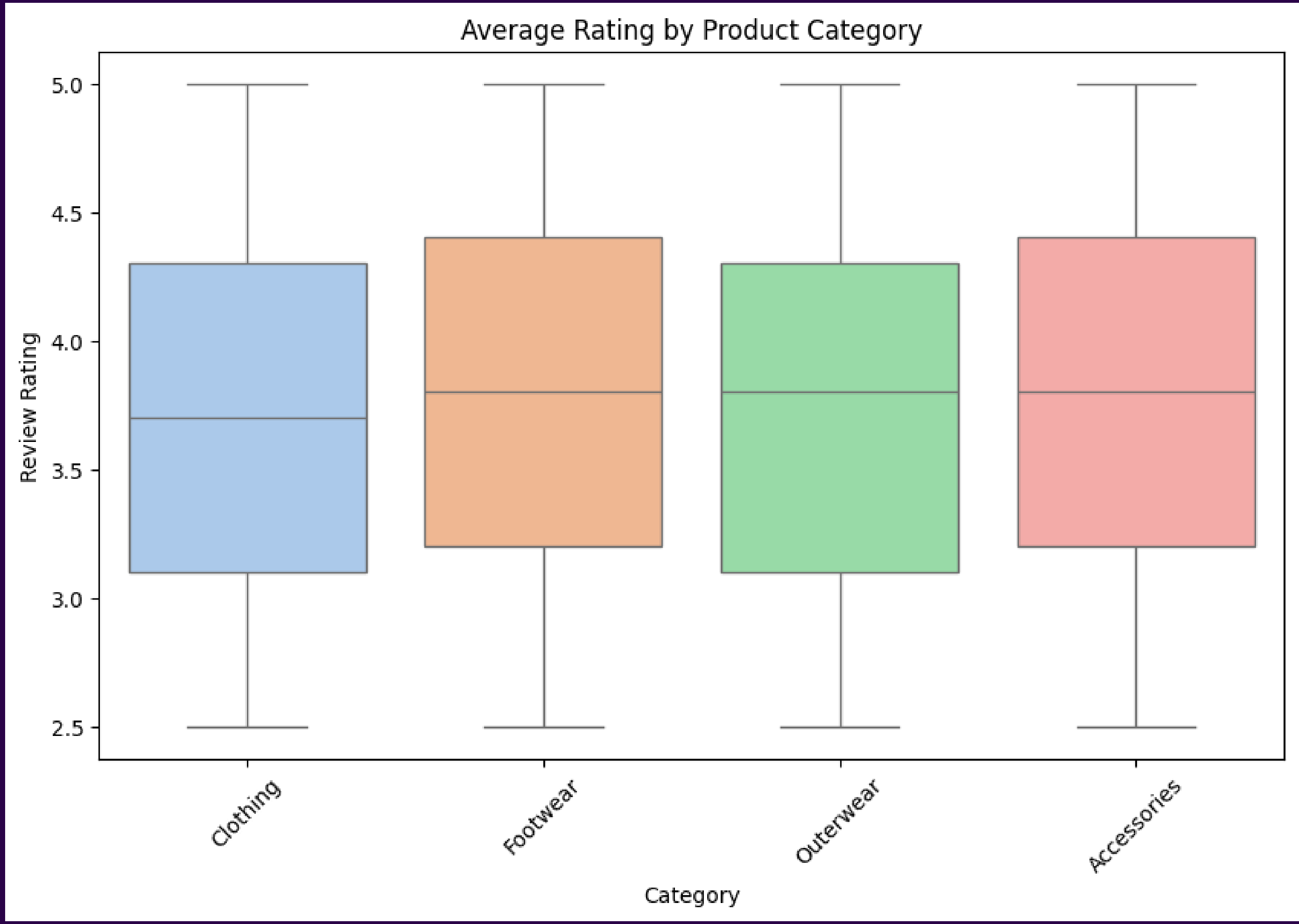
First to market Customers

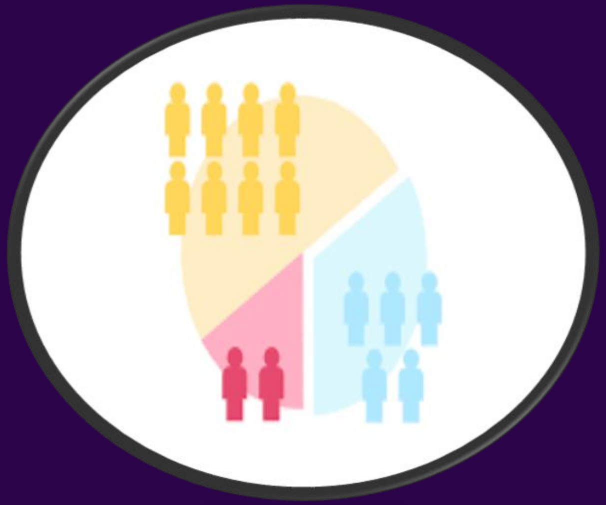
First to market customers gave excellent or average rating to the products they purchased.

Ratings of Loyal Customers

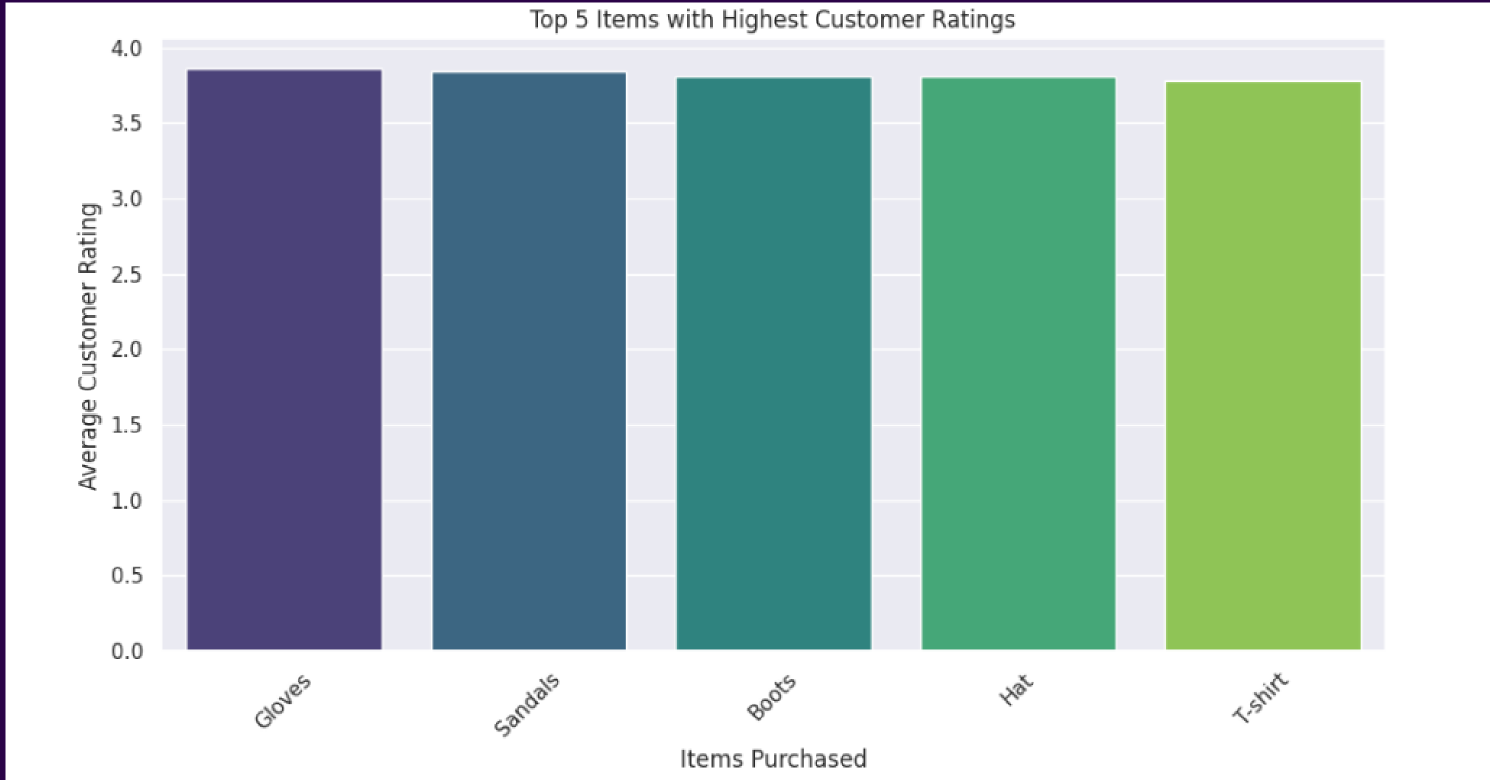
Loyal Customers are not happy about some of the products they are purchasing. They gave **bad** or good rating.

AVERAGE RATINGS FOR THE DIFFERENT PRODUCTS IN SALE


Average Rating by Product Category
Review Rating
5.0
4.5
4.0
3.5
3.0
2.5
Clothing
Footwear
Outerwear
Accessories
Category



TOP 5 PRODUCTS BY CUSTOMER RATING



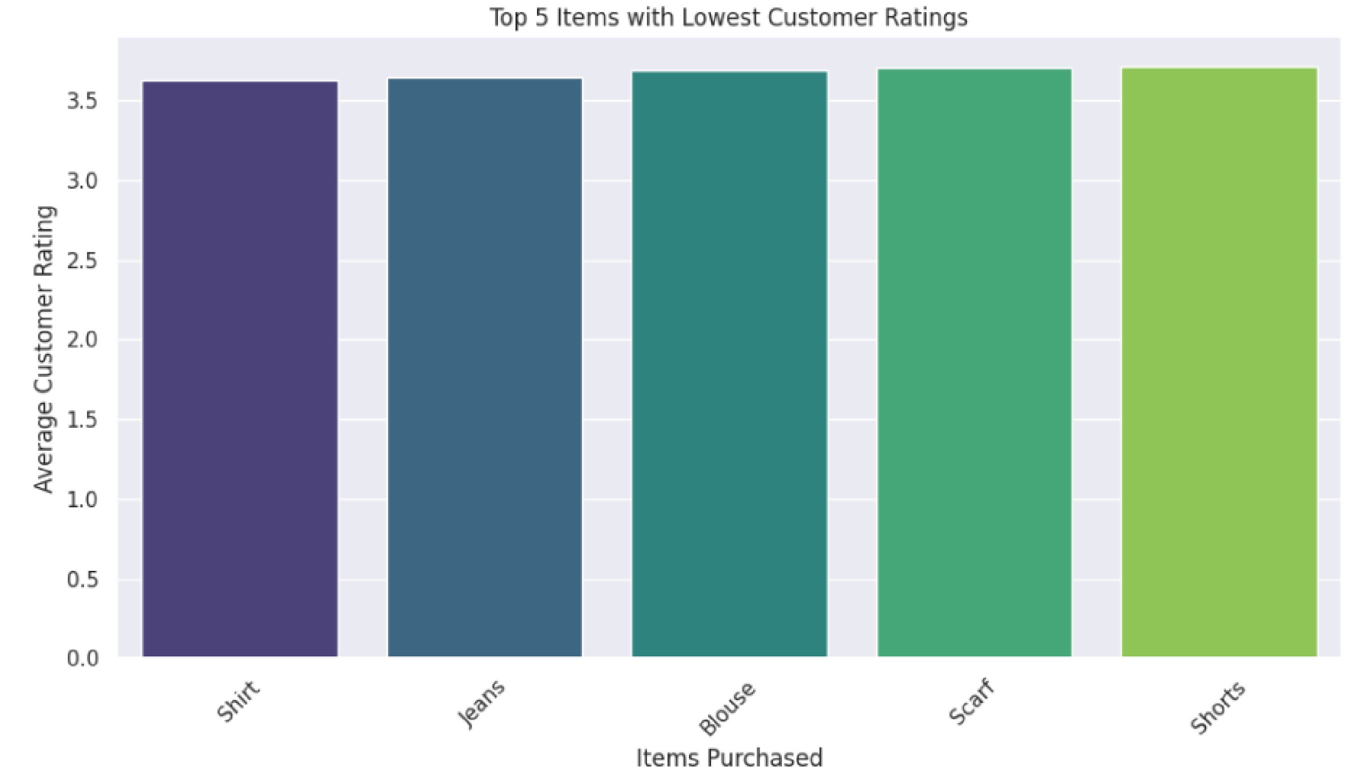
Top 5 items with the lowest average ratings:

	Item Purchased	Review Rating
6	Gloves	3.864286
14	Sandals	3.841250
3	Boots	3.812500
8	Hat	3.806494
24	T-shirt	3.782993

So the Top 5 products in sales are : Gloves, Sandals, Boots, Hat, and T-shirts.

The sale of these items need to be scaled up in order to increase profit in the business model.

PRODUCTS HAVING THE LOWEST CUSTOMER RATINGS



Top 5 items with the lowest average ratings:

	Item Purchased	Review Rating
16	Shirt	3.629586
11	Jeans	3.648387
2	Blouse	3.683626
15	Scarf	3.700000
18	Shorts	3.711465

The items having lowest customer ratings are :

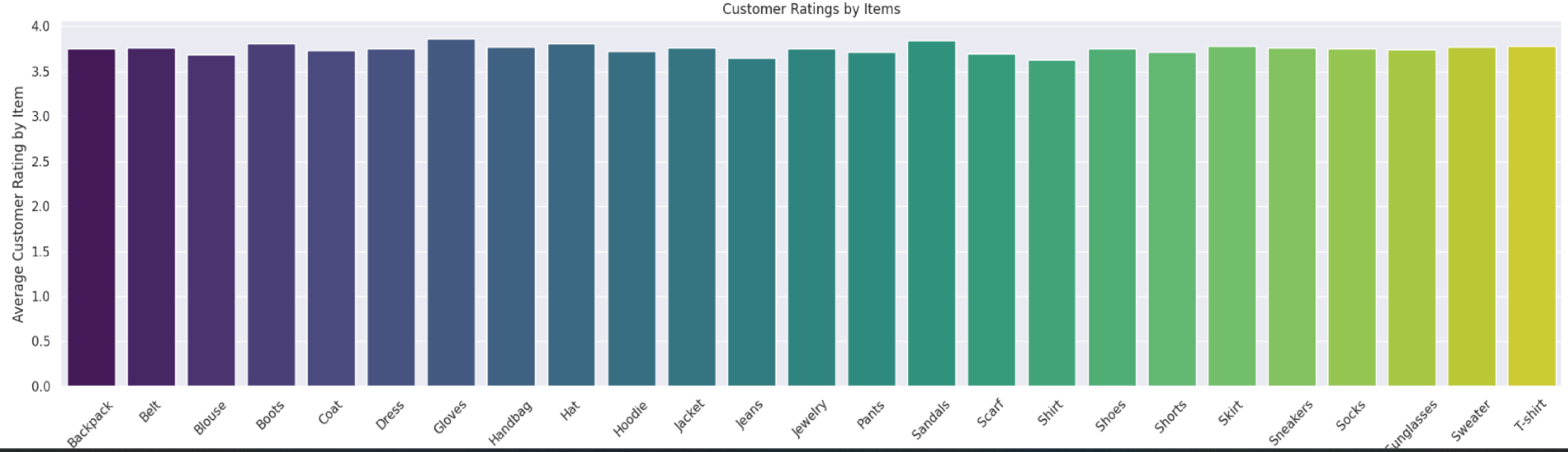
- Jeans
- Blouse
- Scarf
- Shorts
- Shirt (Not T-shirts)

Now poor ratings have been given by loyal customers. This could imply that they were satisfied with these products early on and now, they see a drop in quality.

[103]

```python
avg_ratings = data.groupby('Item Purchased')['Review Rating'].mean().reset_index()


#visualizing the review ratings per Item

plt.figure(figsize=(20, 6))
sns.barplot(x='Item Purchased', y='Review Rating', data=avg_ratings, palette='viridis')
plt.xlabel('Items Purchased')
plt.ylabel('Average Customer Rating by Item')
plt.title('Customer Ratings by Items')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```

```
<ipython-input-103-0017079f4bb4>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Item Purchased', y='Review Rating', data=avg_ratings, palette='viridis')
```



CODE SNIPPETS : CUSTOMER RATINGS BY ITEMS

[104]
```python
# Get the 5 items with the lowest ratings
sorted_ratings = avg_ratings.sort_values(by='Review Rating', ascending=True)
lowest_rated_items = sorted_ratings.head(5)

# Plotting with seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='Item Purchased', y='Review Rating', data=lowest_rated_items, palette='viridis')
plt.xlabel('Items Purchased')
plt.ylabel('Average Customer Rating')
plt.title('Top 5 Items with Lowest Customer Ratings')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Print the 5 items with the lowest ratings
print("Top 5 items with the lowest average ratings:")
print(lowest_rated_items)
```

GRADIENT GEEKS

[105]
```python
# Get the 5 items with the best ratings
sorted_ratings = avg_ratings.sort_values(by='Review Rating', ascending=False)
highest_rated_items = sorted_ratings.head(5)

# Plotting with seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='Item Purchased', y='Review Rating', data=highest_rated_items, palette='viridis')
plt.xlabel('Items Purchased')
plt.ylabel('Average Customer Rating')
plt.title('Top 5 Items with Highest Customer Ratings')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Print the 5 items with the lowest ratings
print("Top 5 items with the lowest average ratings:")
print(highest_rated_items)
```

CODE SNIPPETS : SORTING THE ITEMS ON THE BASIS OF RATINGS

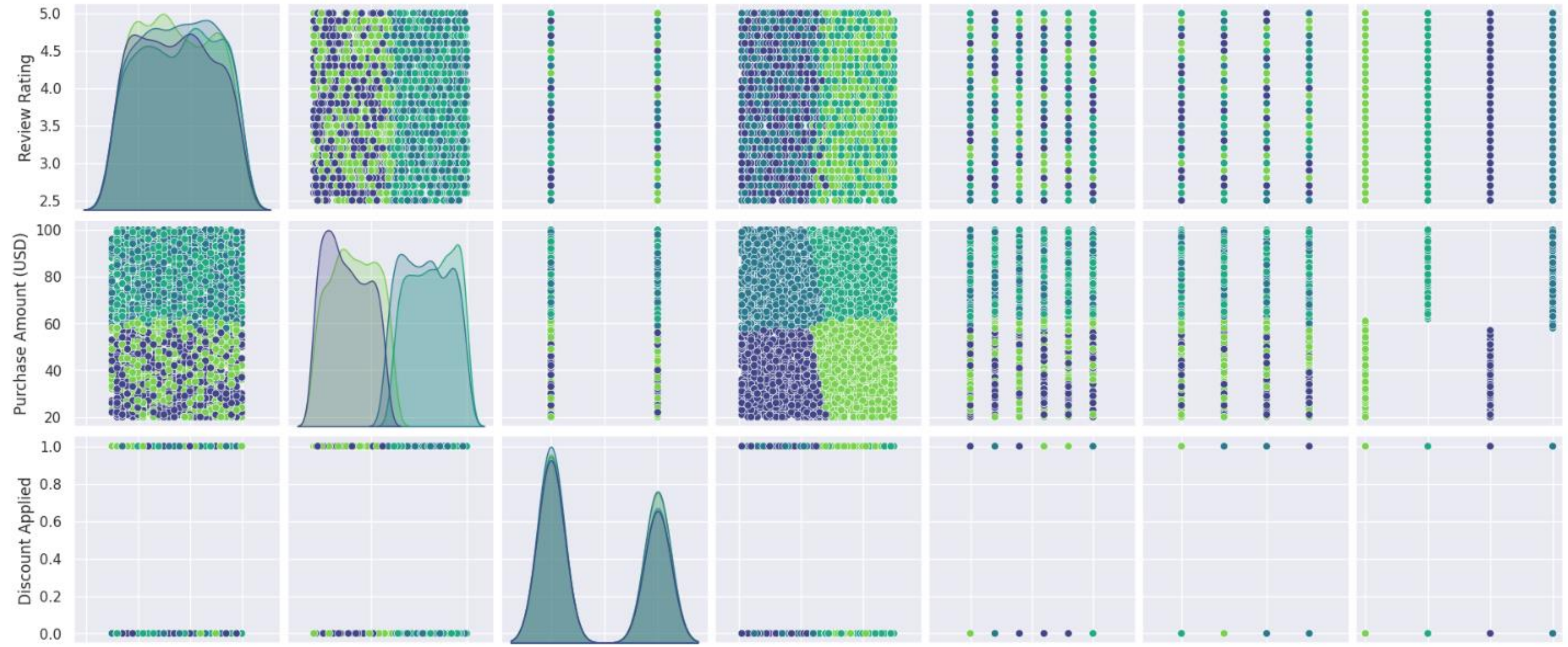
```
[107] from sklearn.cluster import KMeans
      from sklearn.preprocessing import StandardScaler
      from sklearn.preprocessing import LabelEncoder
      import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns

      # Features for customer satisfaction
      features_for_customer_satisfaction = [
          'Review Rating',
          'Purchase Amount (USD)',
          'Discount Applied',
          'Previous Purchases',
          'Shipping Type',
          'Category'
      ]

      satisfaction_data = data[features_for_customer_satisfaction]

      #Encoding the Shipping Type, Category and Discount Applied
      label_encoder = LabelEncoder()
      satisfaction_data['Discount Applied'] = label_encoder.fit_transform(satisfaction_data['Discount Applied'])
      satisfaction_data['Shipping Type'] = label_encoder.fit_transform(satisfaction_data['Shipping Type'])
      satisfaction_data['Category'] = label_encoder.fit_transform(satisfaction_data['Category'])
```

CODE SNIPPETS : ENCODING TEXT FEATURES

```
[108] # Applying KMeans clustering
kmeans = KMeans(n_clusters=4, random_state=42)
satisfaction_data['Cluster'] = kmeans.fit_predict(satisfaction_data)

# Calculate average rating for each cluster
average_ratings = satisfaction_data.groupby('Cluster')['Review Rating'].mean()

# Map cluster names to integers
cluster_names = {
    0: 'Good',
    1: 'Bad',
    2: 'Average',
    3: 'Excellent'
}
satisfaction_data['Cluster Name'] = satisfaction_data['Cluster'].map(cluster_names)

# Visualizing clusters with average ratings
sns.set(style="darkgrid")
plt.figure(figsize=(12, 6))
sns.scatterplot(x='Purchase Amount (USD)', y='Review Rating', hue='Cluster Name', data=satisfaction_data, palette='viridis', legend='full')

for i, txt in enumerate(average_ratings):
    print(f"Cluster {i}: Avg Rating: {round(txt, 2)} ", end="")

plt.title("Review Rating vs. Purchase Amount with Clusters")
plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Review Rating")
plt.legend()
plt.show()
```


CODE SNIPPETS : K MEANS CLUSTERING

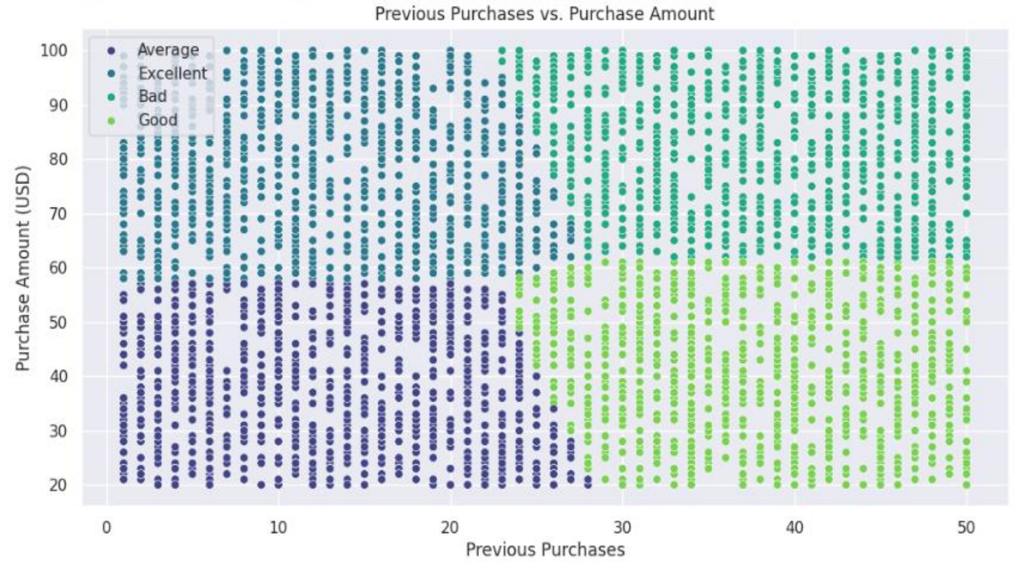
```python
# Visualize clusters
sns.pairplot(satisfaction_data, hue='Cluster Name', palette='viridis')
plt.show()

# Analyzing clusters
# calculating average 'Review Rating' for each cluster
average_ratings = satisfaction_data.groupby('Cluster Name')['Review Rating'].mean()
print("\nAverage Review Ratings by Cluster:")
print(average_ratings)
```



CODE SNIPPETS :USING PAIRPLOT FROM SEABORN

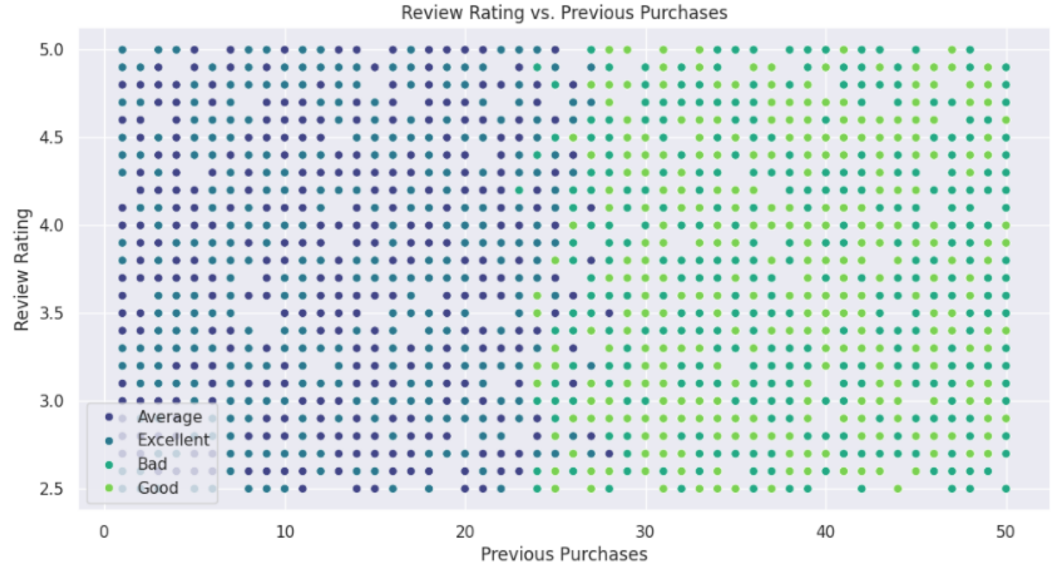
```python
# Visualizing clusters with average ratings
sns.set(style="darkgrid")
plt.figure(figsize=(12, 6))
sns.scatterplot(x='Previous Purchases', y='Purchase Amount (USD)', hue='Cluster Name', data=satisfaction_data, palette='viridis', legend='full')

for i, txt in enumerate(average_ratings):
    print(f"Cluster {i}: Avg Rating: {round(txt, 2)} ", end="")

plt.title("Previous Purchases vs. Purchase Amount")
plt.xlabel("Previous Purchases")
plt.ylabel("Purchase Amount (USD)")
plt.legend()
plt.show()
```

Cluster 0: Avg Rating: 3.72 Cluster 1: Avg Rating: 3.79 Cluster 2: Avg Rating: 3.77 Cluster 3: Avg Rating: 3.72

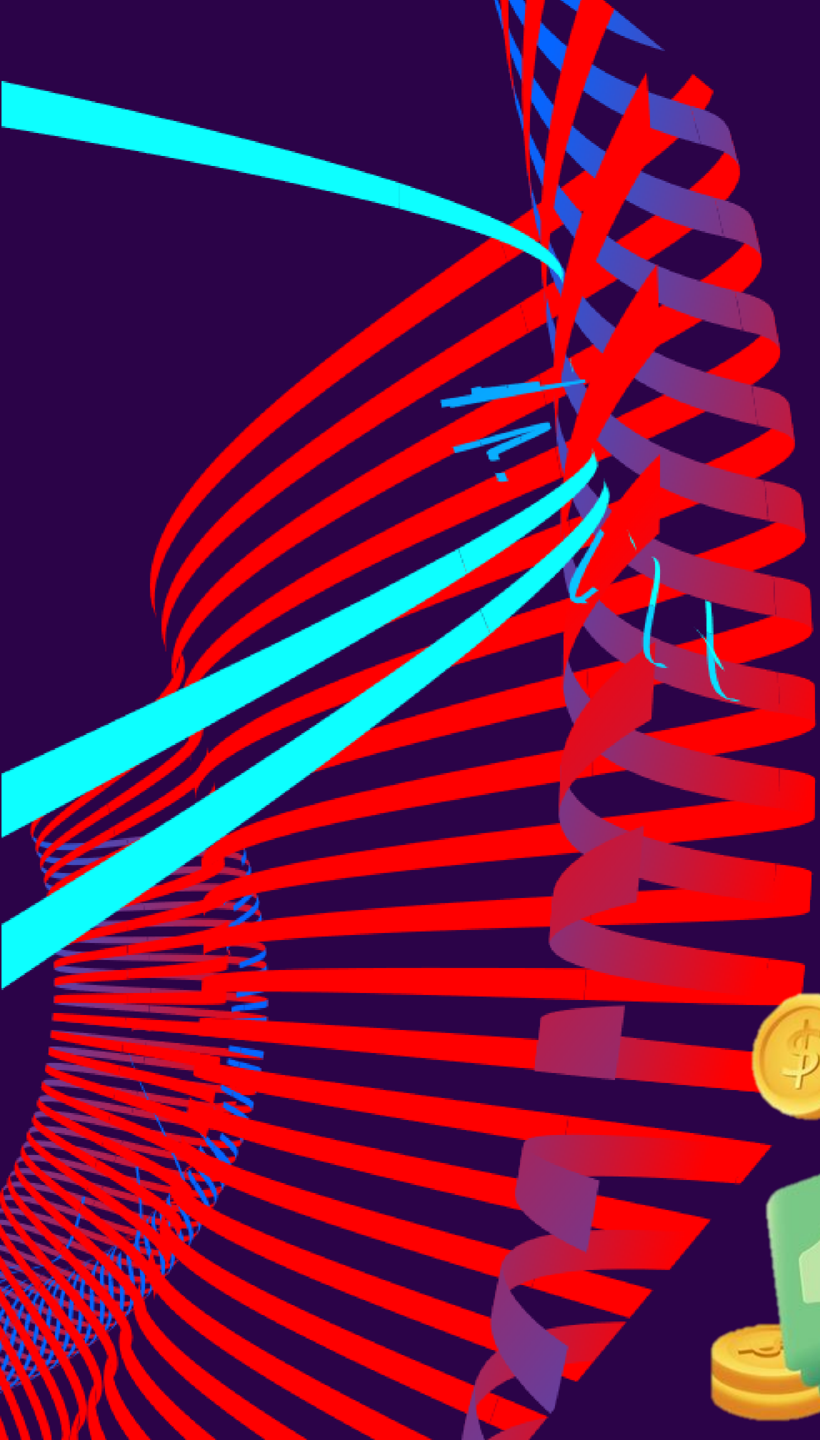



```python
[110] # Visualizing clusters with average ratings
sns.set(style="darkgrid")
plt.figure(figsize=(12, 6))

sns.scatterplot(x='Previous Purchases', y='Review Rating', hue='Cluster Name', data=satisfaction_data, palette='viridis', legend='full')
for i, txt in enumerate(average_ratings):
    print(f"Cluster {i}: Avg Rating: {round(txt, 2)} ", end="")

plt.title("Review Rating vs. Previous Purchases")
plt.xlabel("Previous Purchases")
plt.ylabel("Review Rating")
plt.legend()
plt.show()
```

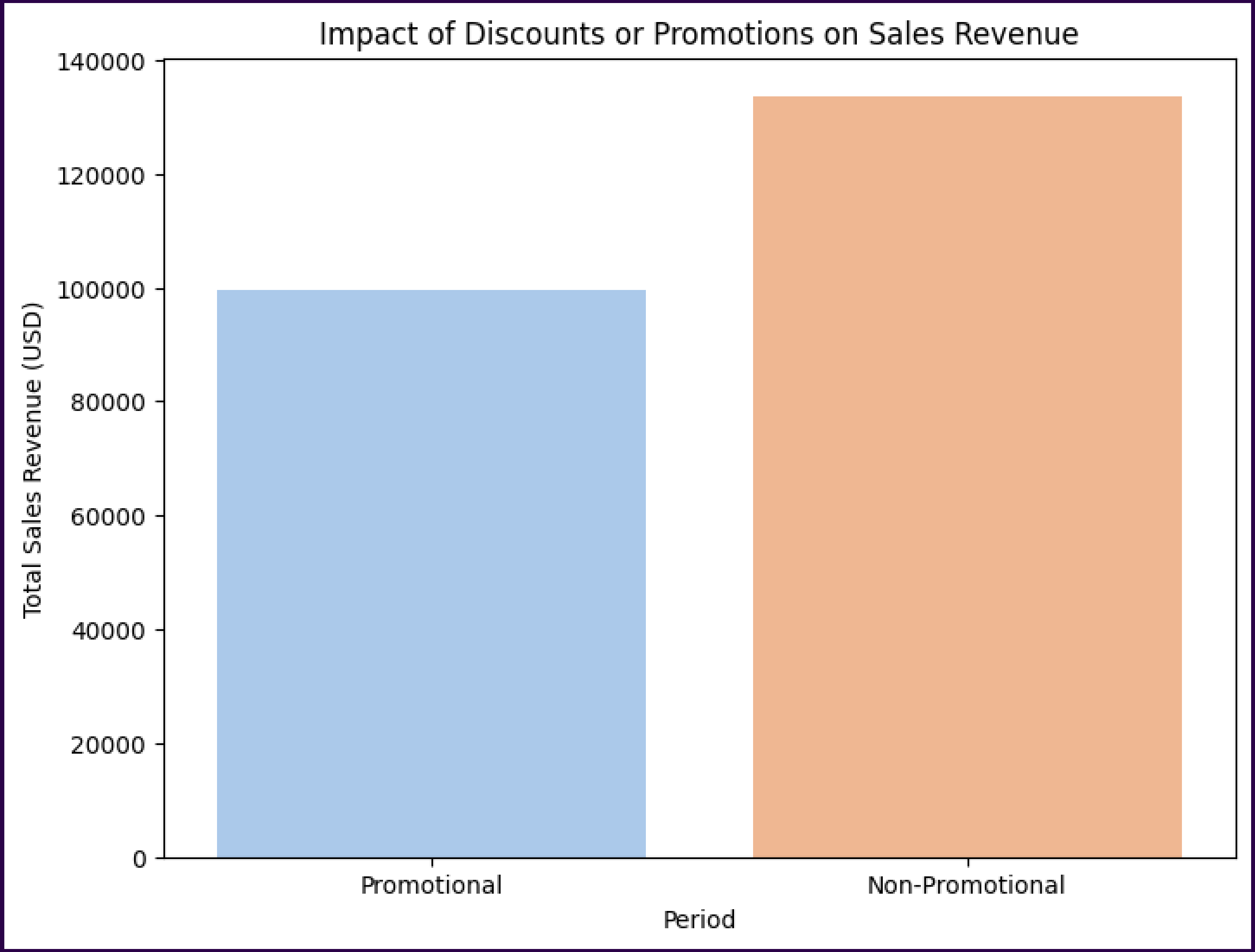
Cluster 0: Avg Rating: 3.72 Cluster 1: Avg Rating: 3.79 Cluster 2: Avg Rating: 3.77 Cluster 3: Avg Rating: 3.72



CODE SNIPPETS : PLOTTING RELEVANT SCATTER PLOTS

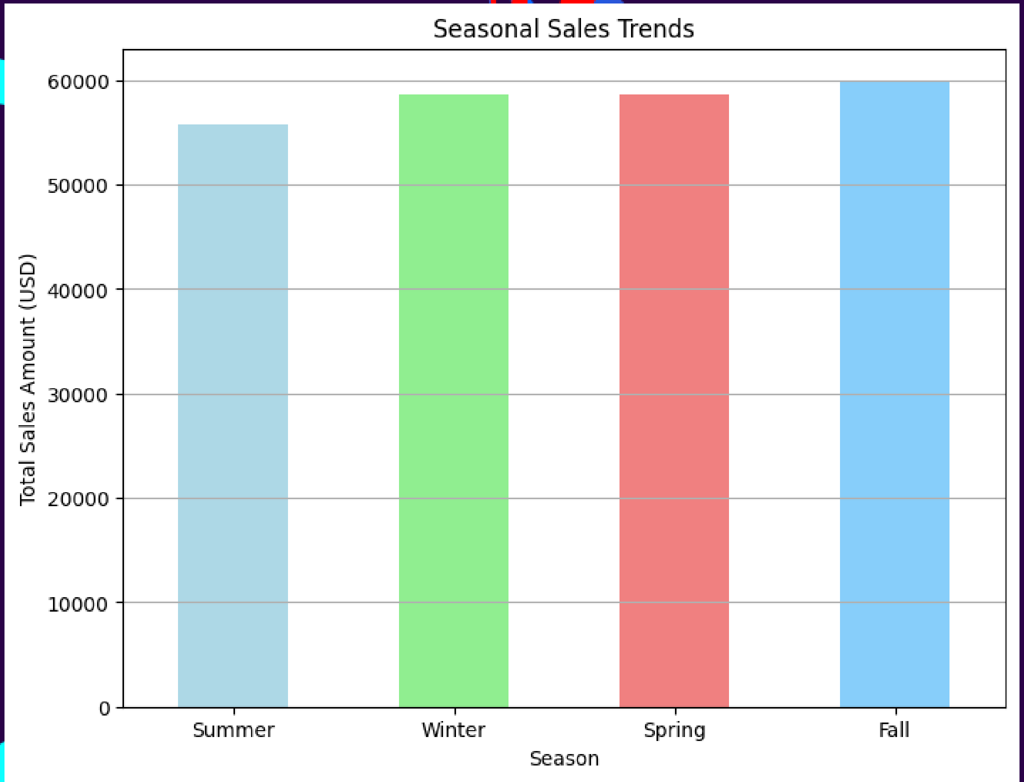


4. IMPACT OF DISCOUNTS OR PROMOTIONS



IMPACT OF DISCOUNTS

IMPACT OF SEASONS



```
seasonal_sales = data.groupby('Season')['Purchase Amount (USD)'].sum()

# 4. Visualize Seasonal Trends in bar chart with different colors in ascending order
plt.figure(figsize=(8, 6))
seasonal_sales.sort_values().plot(kind='bar', color=['lightblue', 'lightgreen', 'lightcoral', 'lightskyblue'])
plt.title('Seasonal Sales Trends')
plt.xlabel('Season')
plt.ylabel('Total Sales Amount (USD)')
plt.xticks(rotation=0)
plt.grid(axis='y')
plt.show()
```

BUSINESS MODEL FOR HIGHER SALES IN FUTURE

Research

Improve the quality of jeans, shirts, shorts, scarfs and blouse.

Abstract

We believe people need more products specifically dedicated to Footwear like boots and sandals and trendy T-shirts.

Implement

Implementing these changes would see a rise in the sales as well as the ratings.

PROBLEM STATEMENT 2:

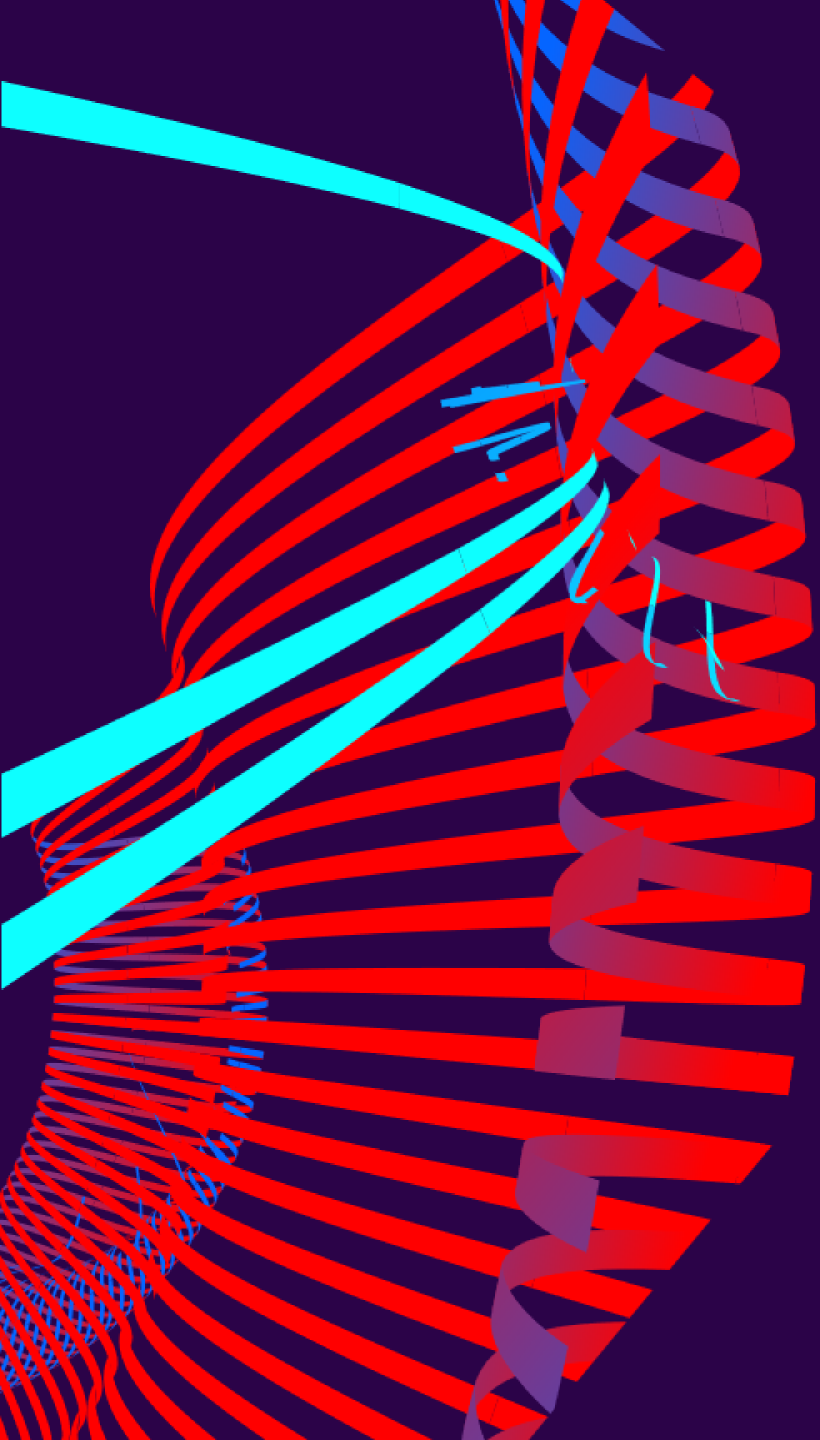
EXTRACTING LEGAL INSIGHTS FROM FIR IMAGES

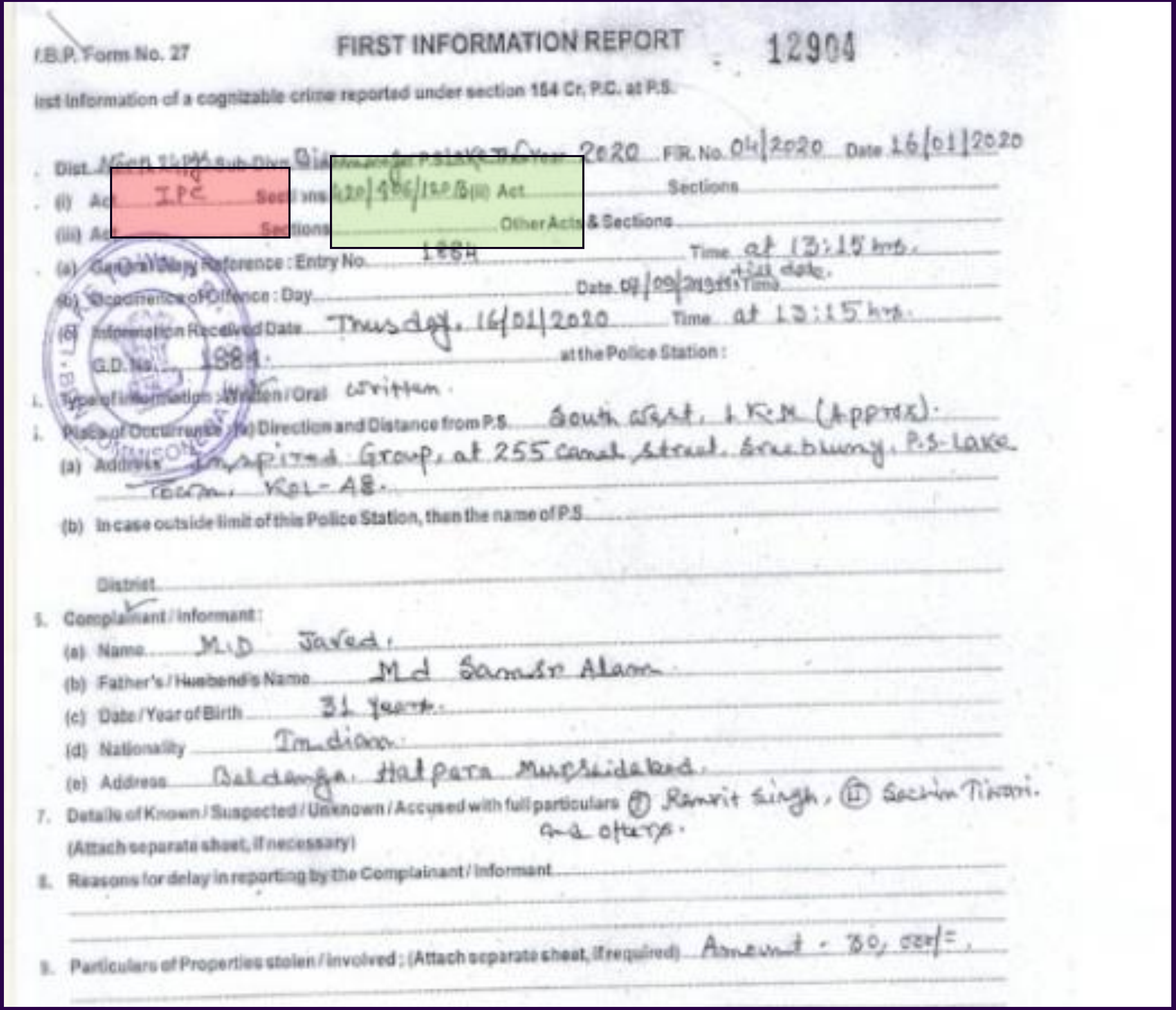
PROBLEM CHALLENGES

1. Develop a state-of-the-art (SOTA) model that can automatically process FIR images and identify the applicable IPC sections:
2. Analyse the textual content within the FIR image to pinpoint the relevant sections of the Indian Penal Code (IPC) associated with the reported crime.
3. Classify the criminal act: Categorise the criminal activity described in the FIR based on the identified IPC sections. Recommend potential punishments: Based on the classified act and relevant IPC sections, suggest the range of punishments typically awarded for such offenses.

OUR IDEA FOR DEVELOPING THE SOTA MODEL

1. Performing Image Segmentation to segregate pixels into groups having similar properties.
2. Performing Object detection to extract the digits from the image of the FIR.
3. Training a Neural Network to perform Handwritten Digit Detection.





I.B.P. Form No. 27 FIRST INFORMATION REPORT 12904

irst Information of a cognizable crime reported under section 154 Cr. P.C. at P.S.

Dist. ... Sub-Divn ... Year 2020 F.I.R. No. 04/2020 Date 16/01/2020

(i) Act IPC Sections 420/406/120B Act Sections

(iii) Act Sections Other Acts & Sections

(a) General Diary Reference : Entry No. 1884 Time at 13:15 hrs.

(b) Occurrence of Offence : Day Date 07/09/2019 Time till date.

(c) Information Received Date Thursday. 16/01/2020 Time at 13:15 hrs.

at the Police Station :

G.D. No. 1884.

Type of information : Written / Oral Written.

Place of Occurrence : (a) Direction and Distance from P.S. South West, 1 K.M. (Approx).

(a) Address Inspired Group, at 255 Canal Street, Sreebhumi, P.S. Lake Town, Kol-48.

(b) In case outside limit of this Police Station, then the name of P.S.

District

5. Complainant / Informant :

(a) Name M.D Javed.

(b) Father's / Husband's Name Md Samsir Alam.

(c) Date / Year of Birth 31 years.

(d) Nationality Indian.

(e) Address Beldanga. Halpara Murshidabad.

7. Details of Known / Suspected / Unknown / Accused with full particulars (i) Ranjit Singh, (ii) Sachin Tiwari. and others.

(Attach separate sheet, if necessary)

8. Reasons for delay in reporting by the Complainant / Informant

9. Particulars of Properties stolen / involved ; (Attach separate sheet, if required) Amount - 30,000/=.

IMAGE SEGMENTATION AND OBJECT DETECTION

DATA PREPROCESSING

Handwritten content adds complexity to text extraction processes. Preprocessing techniques such as image enhancement, noise reduction, and binarization are essential to improve OCR (Optical Character Recognition) accuracy.

OCR TECHNIQUES

Implementing advanced OCR techniques capable of recognizing handwritten text accurately is crucial. This may involve training models specifically for handwritten text recognition or leveraging pre-trained models fine-tuned on diverse datasets.

Flattening our training dataset

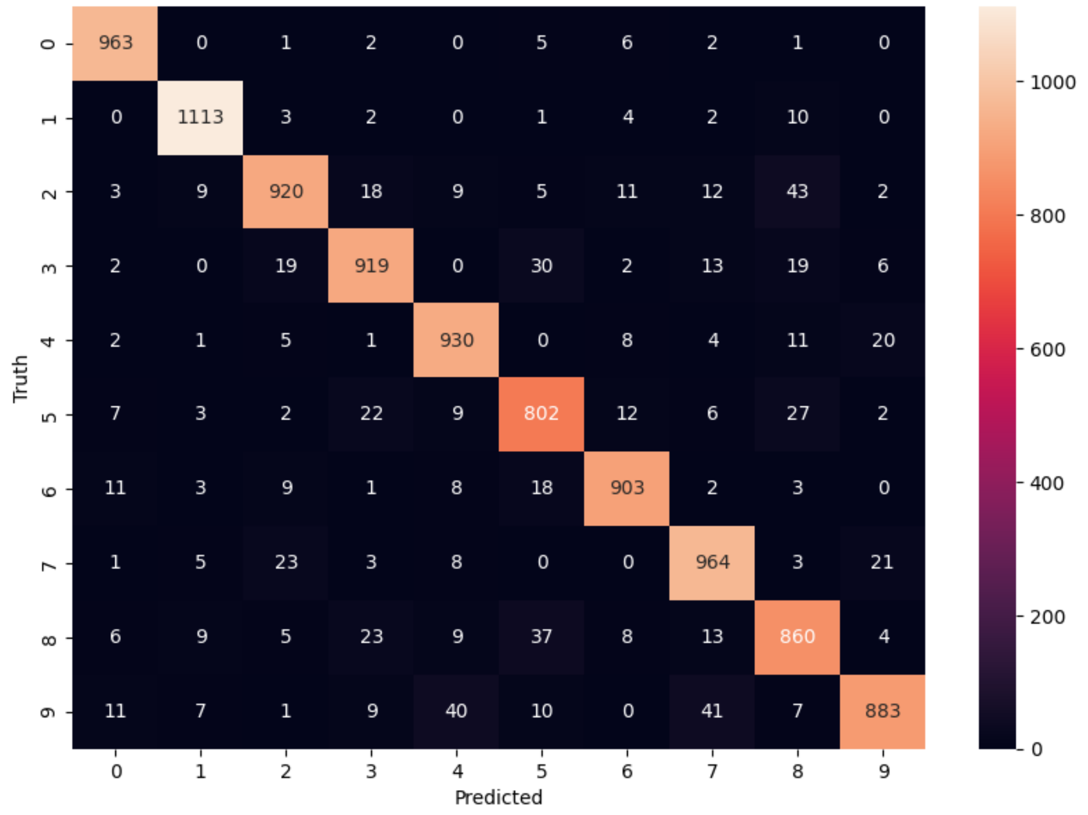
```
[ ]  X_train_flattened = X_train.reshape(len(X_train), 28*28)
     X_test_flattened = X_test.reshape(len(X_test), 28*28)
     X_train_flattened[0]
```

```
model = keras.Sequential([
    keras.layers.Dense(384, input_shape=(784,), activation='tanh'),
    keras.layers.Dense(100, activation='tanh'),
    keras.layers.Dense(35, activation='tanh'),
    keras.layers.Dense(10, activation='sigmoid')
])# 10 output neurons and 784 input neurons

model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

model.fit(X_train_flattened, y_train, epochs=5)
```

```
Epoch 1/5
1875/1875 [==============================] - 14s 7ms/step - loss: 0.2485 - accuracy: 0.9286
Epoch 2/5
1875/1875 [==============================] - 18s 10ms/step - loss: 0.1186 - accuracy: 0.9638
Epoch 3/5
1875/1875 [==============================] - 26s 14ms/step - loss: 0.0894 - accuracy: 0.9733
Epoch 4/5
1875/1875 [==============================] - 14s 7ms/step - loss: 0.0688 - accuracy: 0.9793
Epoch 5/5
1875/1875 [==============================] - 14s 7ms/step - loss: 0.0584 - accuracy: 0.9812
<keras.src.callbacks.History at 0x798b7bae8a90>
```

```
import seaborn as sn
plt.figure(figsize = (10, 7))
sn.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(95.72222222222221, 0.5, 'Truth')
```



HANDWRITTEN DIGIT RECOGNITION USING NEURAL NETWORKS

LAYOUT VARIATION

FIR documents may vary in layout, font size, and formatting.

Developing models resilient to layout variations requires robust feature extraction techniques and flexible architectures capable of handling diverse input formats.

TEXT ANALYSIS AND INFORMATION EXTRACTION

- Utilization Natural Language Processing (NLP) techniques to analyze the extracted text and identify relevant legal information, such as IPC sections.
- Development of models for named entity recognition (NER) and semantic role labeling (SRL) to extract actionable insights from the textual content of FIRs.

CRIME CLASSIFICATION

- To build robust models for classifying the criminal activity described in FIRs based on the identified IPC sections.
- Utilization of machine learning or deep learning algorithms to categorize crimes into predefined classes or categories.

PUNISHMENT PREDICTION

- Recommendation of potential punishments based on the classified criminal act and relevant IPC sections.
- Development of models to predict the range of punishments typically awarded for specific offenses, considering legal precedents and sentencing guidelines.

MODEL EVALUATION AND VALIDATION

•Evaluation of the performance of the developed models using appropriate metrics such as accuracy, precision, recall, and F1-score.

•Validation of the models on a separate test dataset to ensure generalization and robustness across different FIR documents and legal scenarios.

OUR TECH STACK

seaborn

python

NumPy

Google Colaboratory

matplotlib

OUR TEAM

UTTAM

MAHATA

SIDDHARTH

SEN

ANURAG
GHOSH

SUCHANA
HAZRA

THANK YOU