EDS Mini Project:

Group members name: Siddhi Kiwale(673)

Siddhi Shinde(682)

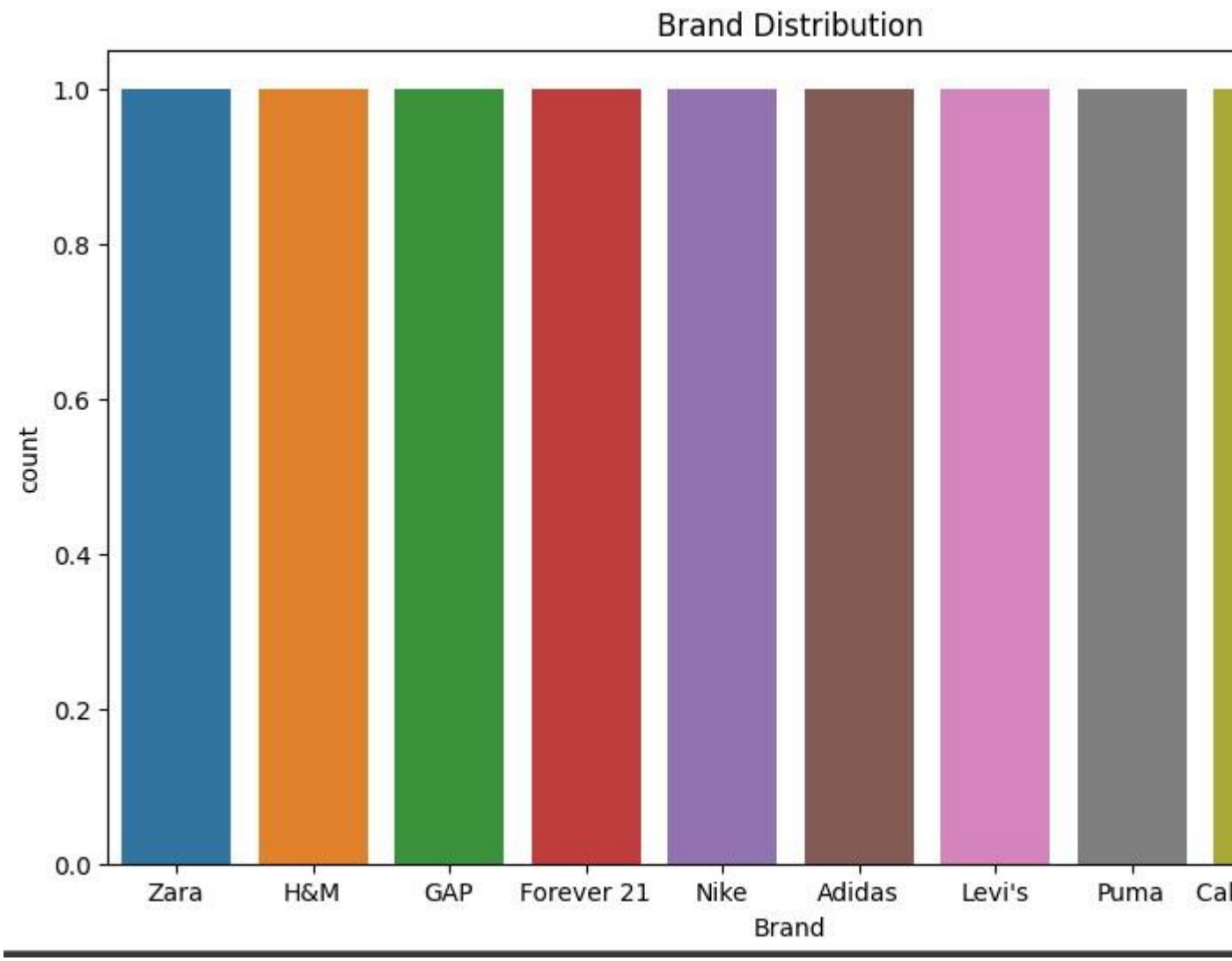
Rutuja Wanwe(668)

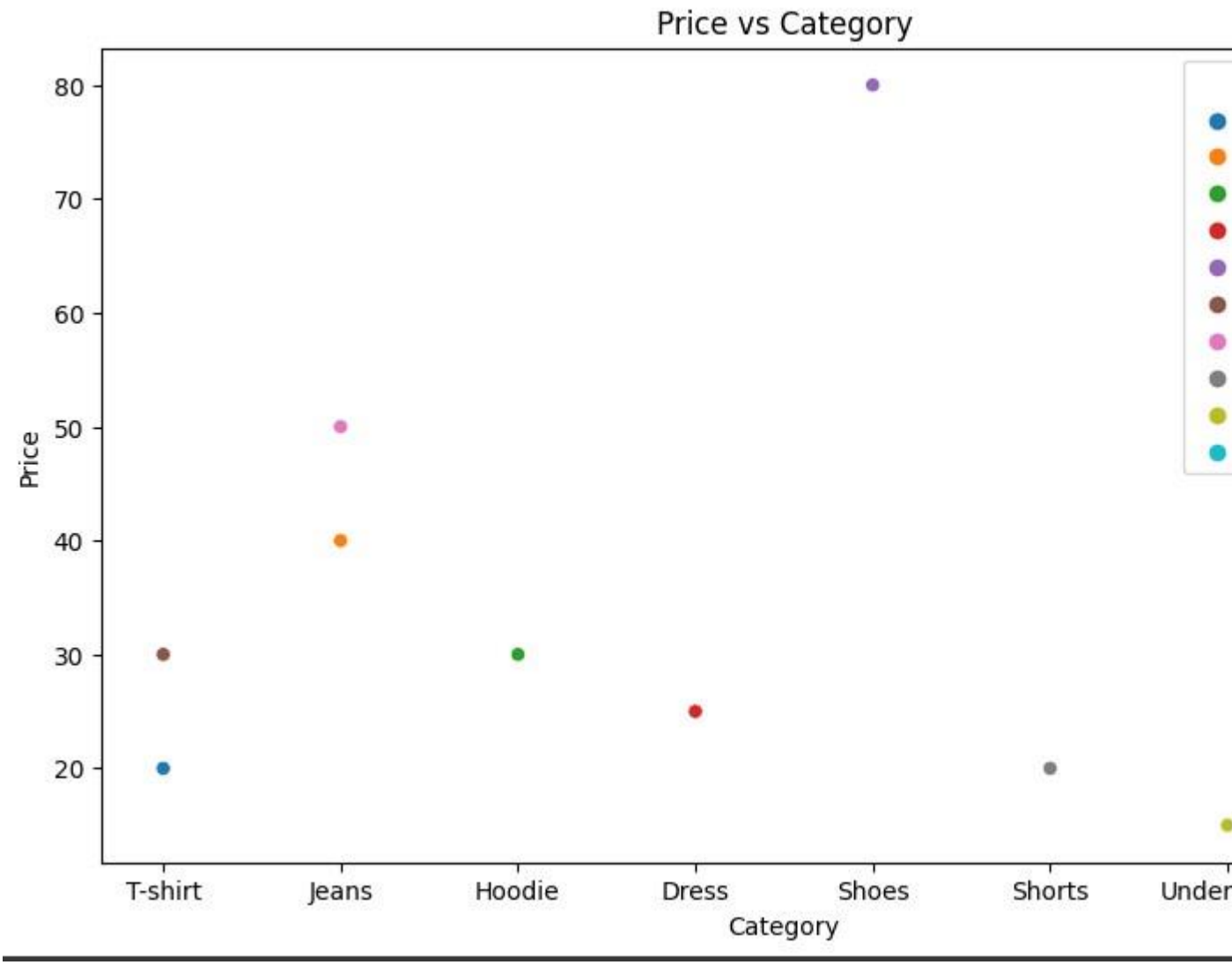
Problem Statement: Implement a mini project based on classification (Linear Regression / KNN Classification) or Clustering (K-Means) and also Develop an interactive dashboard using the matplotlib/Seaborn library.

Data set:

```
#Develop an interactive dashboard using the matplotlib/Seaborn library.

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data_set=pd.read_csv('/content/sample_data/colthing brand.csv')
data_set

# Bar chart - Brand Distribution
plt.figure(figsize=(10, 6))
sns.countplot(data=data_set, x='Brand')
plt.title('Brand Distribution')

# Scatter plot - Price vs Category
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data_set, x='Category', y='Price', hue='Brand')
plt.title('Price vs Category')

plt.show()
```

output:


Brand Distribution
1.0
0.8
0.6
0.4
0.2
0.0
count
Zara
H&M
GAP
Forever 21
Nike
Adidas
Levi's
Puma
Brand




```
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the dataset
data_set=pd.read_csv('/content/sample_data/colthing brand.csv')
data_set

df = pd.DataFrame(data_set)

# Splitting the dataset into training and test sets
X = df[['Brand', 'Category', 'Color', 'Size']]
y = df['Price']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Printing the shapes of the training and test sets
```

```
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

output:

```
X_train shape: (7, 4)
X_test shape: (3, 4)
y_train shape: (7,)
y_test shape: (3,)
```

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
# Encode categorical variables
label_encoder = LabelEncoder()
df['Brand'] = label_encoder.fit_transform(df['Brand'])
df['Category'] = label_encoder.fit_transform(df['Category'])
df['Color'] = label_encoder.fit_transform(df['Color'])
df['Size'] = label_encoder.fit_transform(df['Size'])

# Split the dataset into training and test sets
X = df[['Brand', 'Category', 'Color', 'Size']]
y = df['Price']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Fit the Simple Linear Regression model to the training dataset
regression_model = LinearRegression()
regression_model.fit(X_train, y_train)

# Print the coefficients and intercept of the model
print("Coefficients:", regression_model.coef_)
print("Intercept:", regression_model.intercept_)
```

output:

```
Coefficients: [ -4.71367004   9.00917332   7.12626898 -11.45934384]
Intercept: 52.873951437848
```

```
# Predict the results for training and test sets
```

```
y_train_pred = regression_model.predict(X_train)
y_test_pred = regression_model.predict(X_test)

# Print the predicted results
print("Training set predictions:", y_train_pred)
print("Test set predictions:", y_test_pred)
```

```
Training set predictions: [18.66858565 21.87994654 39.07596493 36.31915549 77.67435656 18.
 47.32394788]
Test set predictions: [76.84646313 24.78585539 91.13657267]
```