# EDS Mini Project -(G Division)

**Group members name:** Kanishka Garud – 723

Apurva Koli - 736

## Problem Statement:

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

**Data set:**

| A | B | C | D | E |
|---|---|---|---|---|
| Brand | Category | Price | Color | Size |
| Zara | T-shirt | 19.99 | Black | S |
| H&M | Jeans | 39.99 | Blue | M |
| GAP | Hoodie | 29.99 | Gray | L |
| Forever 2 | Dress | 24.99 | Red | S |
| Nike | Shoes | 79.99 | White | 8 |
| Adidas | T-shirt | 29.99 | Blue | L |
| Levi's | Jeans | 49.99 | Black | 32 |
| Puma | Shorts | 19.99 | Gray | XL |
| Calvin Kle | Underwea | 14.99 | Black | M |
| Tommy Hi | Shirt | 34.99 | White | M |

**Code:**

```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
data_set=pd.read_csv('/content/sample_data/colthing brand.csv')
data_set
```

| | Brand | Category | Price | Color | Size |
|---|---|---|---|---|---|
| 0 | Zara | T-shirt | 19.99 | Black | S |
| 1 | H&M | Jeans | 39.99 | Blue | M |
| 2 | GAP | Hoodie | 29.99 | Gray | L |
| 3 | Forever 21 | Dress | 24.99 | Red | S |
| 4 | Nike | Shoes | 79.99 | White | 8 |
| 5 | Adidas | T-shirt | 29.99 | Blue | L |
| 6 | Levi's | Jeans | 49.99 | Black | 32 |
| 7 | Puma | Shorts | 19.99 | Gray | XL |
| 8 | Calvin Klein | Underwear | 14.99 | Black | M |
| 9 | Tommy Hilfiger | Shirt | 34.99 | White | M |

```
x=data_set.iloc[:,:-1].values
y=data_set.iloc[:,1].values
#splitting the dataset into training and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train,
y_test=train_test_split(x,y,test_size=1/3,random_state=0)
print(x_train)
```

```
[['H&M' 'Jeans' 39.99 'Blue']
 ["Levi's" 'Jeans' 49.99 'Black']
 ['Puma' 'Shorts' 19.99 'Gray']
 ['Forever 21' 'Dress' 24.99 'Red']
 ['Zara' 'T-shirt' 19.99 'Black']
 ['Adidas' 'T-shirt' 29.99 'Blue']]
```

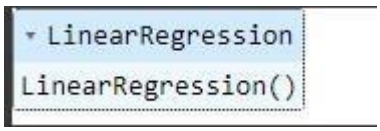
# Code: Linear Regression

```
#linear regression
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
data_set=pd.read_csv('/content/sample_data/colthing_brand.csv')
```

```
data_set

df = pd.DataFrame(data_set)
# Create a linear regression object
model = LinearRegression()

model.fit(X_train, y_train)
```

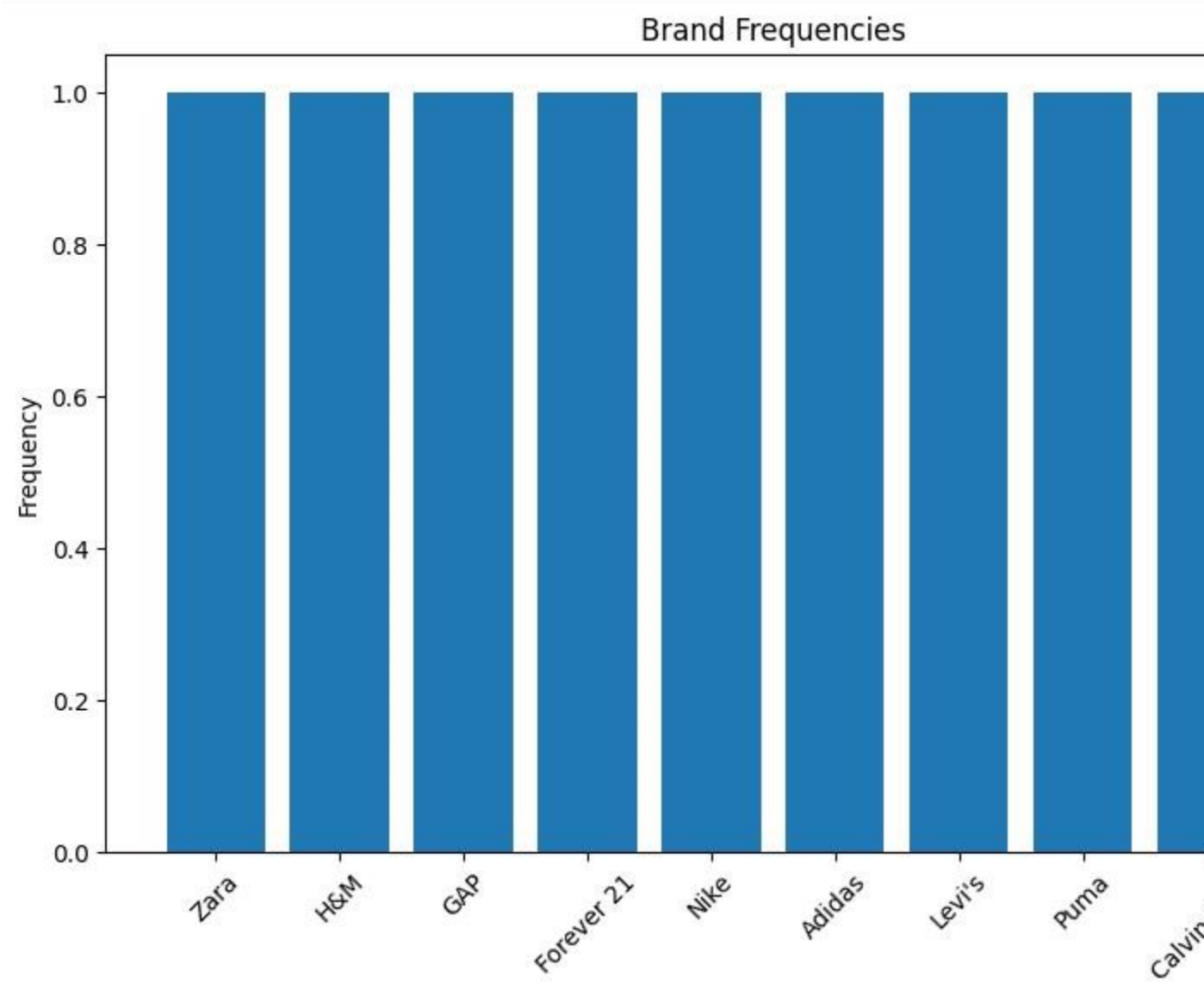



```
#print the coefficient
print("Intercept:", model.intercept_) # Intercept term
print("Coefficients:", model.coef_)  # Coefficients for each feature
```

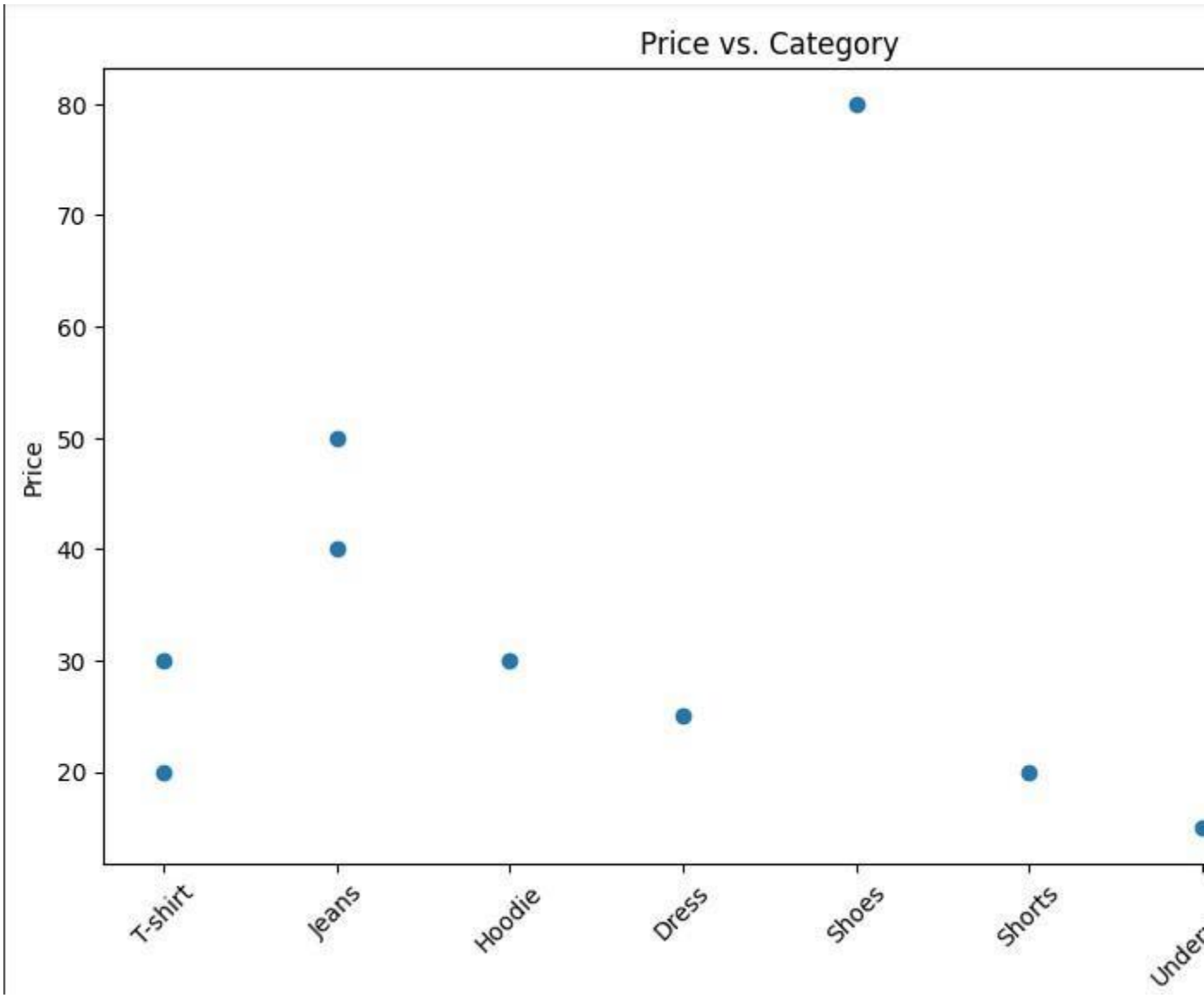
```
Intercept: 7.105427357601002e-15
Coefficients: [ 8.09593144e-16  1.66533454e-15  1.00000000e+00  7.35522754e-16
 -2.59514632e-15]
```

# Code:Visualization

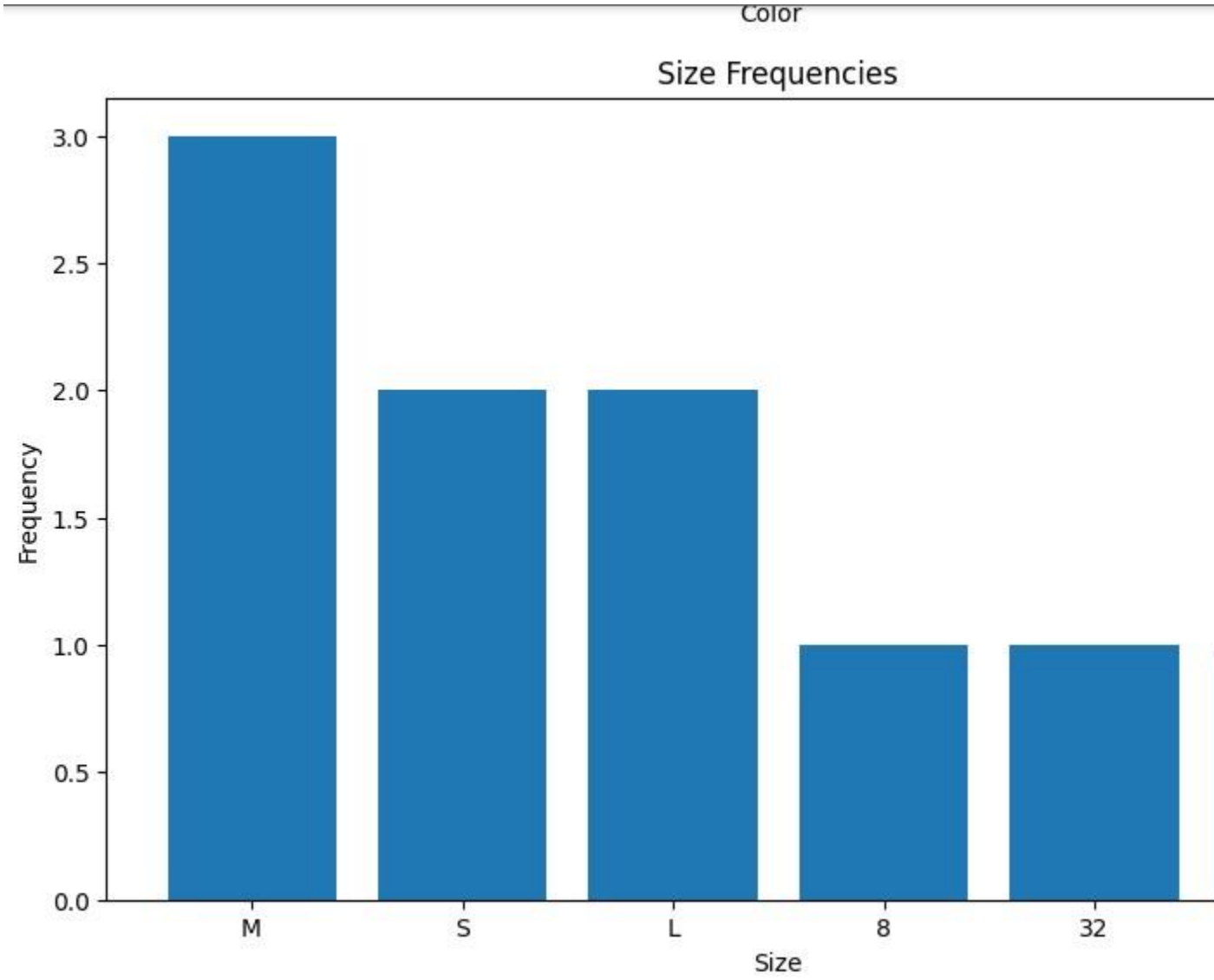
```
#visaulization
#bar plot of brand frequencies
brand_counts = df['Brand'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(brand_counts.index, brand_counts.values)
plt.xlabel('Brand')
plt.ylabel('Frequency')
plt.title('Brand Frequencies')
plt.xticks(rotation=45)
plt.show()
```




```
# Scatter plot of price vs. category
plt.figure(figsize=(10, 6))
plt.scatter(df['Category'], df['Price'])
plt.xlabel('Category')
plt.ylabel('Price')
plt.title('Price vs. Category')
plt.xticks(rotation=45)
plt.show()
```




```
# Bar plot of size frequencies
size_counts = df['Size'].value_counts()
plt.figure(figsize=(10, 6))
plt.bar(size_counts.index, size_counts.values)
plt.xlabel('Size')
plt.ylabel('Frequency')
plt.title('Size Frequencies')
plt.show()
```




## Code: Manupulation

```
#manupulation
import pandas as pd

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

df = pd.DataFrame(data_set)

# Select specific columns
selected_columns = df[['Brand', 'Price']]
print(selected columns)
```

| | Brand | Price |
|---|---|---|
| 0 | Zara | 19.99 |
| 1 | H&M | 39.99 |
| 2 | GAP | 29.99 |
| 3 | Forever 21 | 24.99 |
| 4 | Nike | 79.99 |
| 5 | Adidas | 29.99 |
| 6 | Levi's | 49.99 |
| 7 | Puma | 19.99 |
| 8 | Calvin Klein | 14.99 |
| 9 | Tommy Hilfiger | 34.99 |

```
# Filter rows based on conditions
filtered_rows = df[df['Price'] > 30]
print(filtered_rows)
```

| | Brand | Category | Price | Color | Size |
|---|---|---|---|---|---|
| 1 | H&M | Jeans | 39.99 | Blue | M |
| 4 | Nike | Shoes | 79.99 | White | 8 |
| 6 | Levi's | Jeans | 49.99 | Black | 32 |
| 9 | Tommy Hilfiger | Shirt | 34.99 | White | M |

```
# Sort the dataframe by a column
sorted_df = df.sort_values('Price', ascending=False)
print(sorted_df)
```

| | Brand | Category | Price | Color | Size |
|---|---|---|---|---|---|
| 4 | Nike | Shoes | 79.99 | White | 8 |
| 6 | Levi's | Jeans | 49.99 | Black | 32 |
| 1 | H&M | Jeans | 39.99 | Blue | M |
| 9 | Tommy Hilfiger | Shirt | 34.99 | White | M |
| 2 | GAP | Hoodie | 29.99 | Gray | L |
| 5 | Adidas | T-shirt | 29.99 | Blue | L |
| 3 | Forever 21 | Dress | 24.99 | Red | S |
| 0 | Zara | T-shirt | 19.99 | Black | S |
| 7 | Puma | Shorts | 19.99 | Gray | XL |
| 8 | Calvin Klein | Underwear | 14.99 | Black | M |

```
# Group data and calculate statistics
grouped_data = df.groupby('Category').mean()
print(grouped_data)
```

| Category | Price |
|---|---|
| Dress | 24.99 |
| Hoodie | 29.99 |
| Jeans | 44.99 |
| Shirt | 34.99 |
| Shoes | 79.99 |
| Shorts | 19.99 |
| T-shirt | 24.99 |
| Underwear | 14.99 |

```
# Remove duplicate rows
df = df.drop_duplicates()
print(df)
```

| | Brand | Category | Price | Color | Size |
|---|---|---|---|---|---|
| 0 | Zara | T-shirt | 19.99 | Black | S |
| 1 | H&M | Jeans | 39.99 | Blue | M |
| 2 | GAP | Hoodie | 29.99 | Gray | L |
| 3 | Forever 21 | Dress | 24.99 | Red | S |
| 4 | Nike | Shoes | 79.99 | White | 8 |
| 5 | Adidas | T-shirt | 29.99 | Blue | L |
| 6 | Levi's | Jeans | 49.99 | Black | 32 |
| 7 | Puma | Shorts | 19.99 | Gray | XL |
| 8 | Calvin Klein | Underwear | 14.99 | Black | M |
| 9 | Tommy Hilfiger | Shirt | 34.99 | White | M |

## Code: K-means clustering

```
#k-means clustering
# Preprocess categorical variables
label_encoders = {}
categorical_cols = ['Brand', 'Category', 'Color', 'Size']

for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    df[col] = label_encoders[col].fit_transform(df[col])

# Select the features for clustering
X = df[['Price', 'Color', 'Size']]

# Perform K-Means Clustering
k = 3  # Number of clusters
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(X)

# Get the cluster labels for each data point
cluster_labels = kmeans.labels_
```

```
# Add the cluster labels to the dataframe
df['Cluster'] = cluster_labels

# Print the cluster labels for each data point
print(df[['Brand', 'Cluster']])

# Get the cluster centers
cluster_centers = kmeans.cluster_centers_

# Print the cluster centers
print("Cluster Centers:")
for i, center in enumerate(cluster_centers):
    print("Cluster", i+1, "Center:", center)
```

```
   Brand  Cluster
0      9        1
1      4        0
2      3        1
3      2        1
4      6        2
5      0        1
6      5        0
7      7        1
8      1        1
9      8        0
Cluster Centers:
Cluster 1 Center: [41.65666667  1.66666667  2.        ]
Cluster 2 Center: [23.32333333  1.33333333  3.33333333]
Cluster 3 Center: [79.99  4.    1.  ]
```

# Code: KNN clasification

```
#KNN clasification
# Preprocess categorical variables
label_encoders = {}
categorical_cols = ['Brand', 'Category', 'Color', 'Size']

for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    df[col] = label_encoders[col].fit_transform(df[col])

# Select features and target variable
X = df[['Price', 'Color', 'Size']]
y = df['Brand']
# Split the data into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Perform K-NN Classification
k = 3  # Number of neighbors
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
# Predict on the test set
y_pred = knn.predict(X_test)
# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.0
```