# Brand Recommendation System Using Collaborative Filtering

## Brand Recommendation System Using Collaborative Filtering

## 1. Introduction

The primary objective of this project was to develop a brand recommendation system using collaborative filtering techniques. The system aims to suggest products based on user preferences and ratings, thereby enhancing the shopping experience in a fashion product context. The dataset utilized for this project consists of various fashion products along with user ratings, price, and brand information.

## 2. Methodology

### 2.1 Data Preparation

1. **Data Loading:**
   The dataset was loaded using the Pandas library. The initial examination of the dataset included checking for null values and duplicate entries. This is critical to ensure the quality of the data used for training the recommendation model.

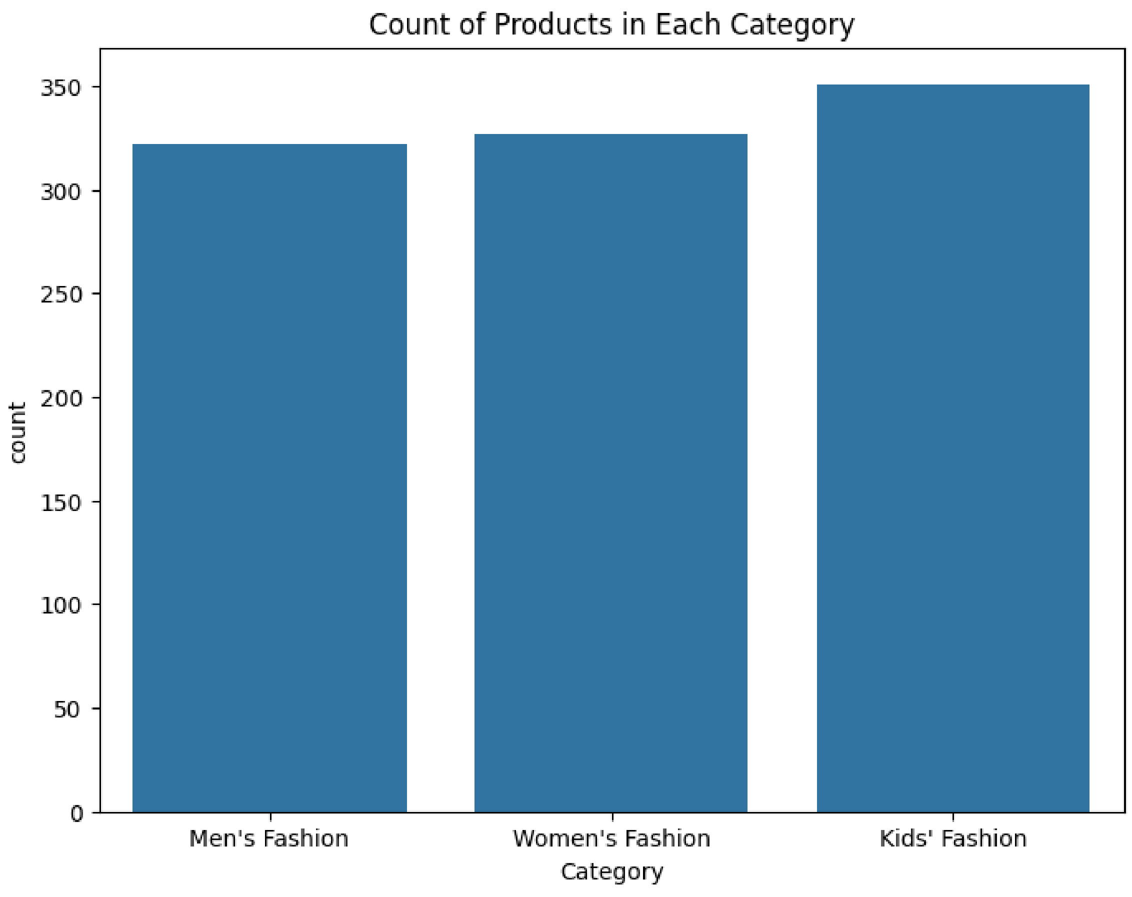
   ```
   df = pd.read_csv("/content/fashion_products.csv")
   df.isnull().sum() # no null values
   df.duplicated().sum()# no duplicated values
   ```

2. **Data Visualization:**

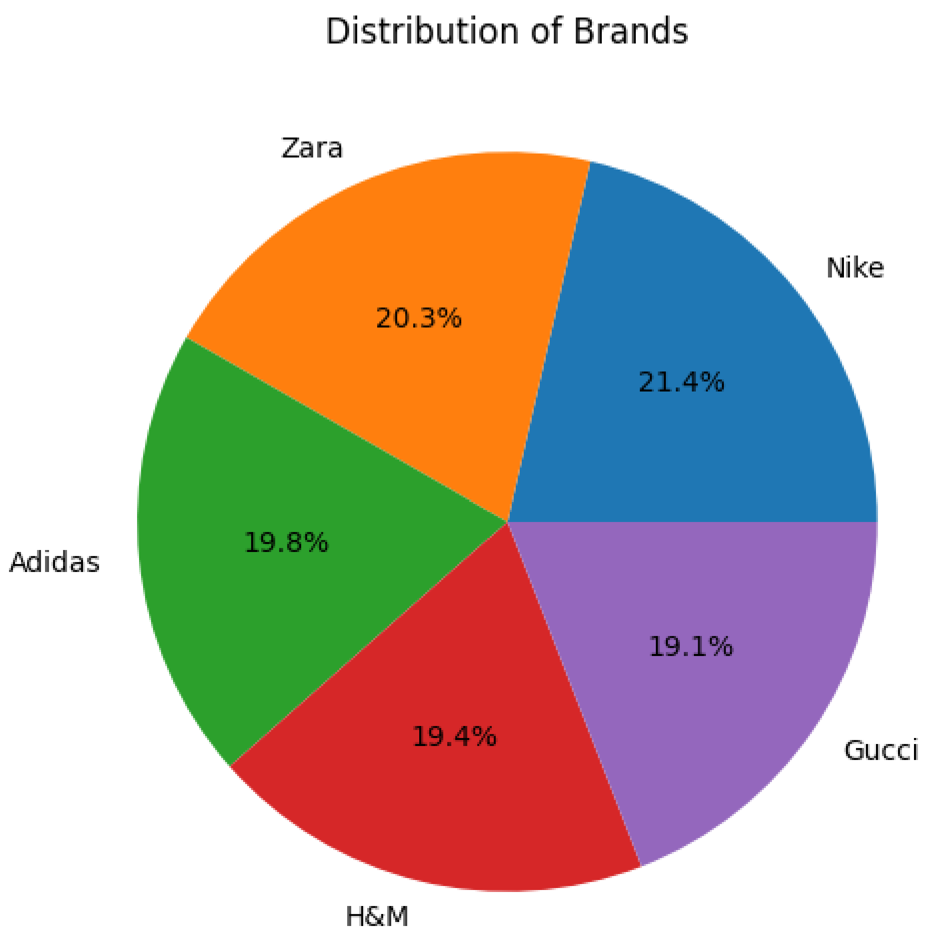
   - A count plot was generated to visualize the distribution of products across different categories.

- A pie chart was created to display the distribution of brands, helping to understand the product diversity.

```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Category')
plt.title('Count of Products in Each Category')
plt.show()
```



```
plt.figure(figsize=(8, 6))
df['Brand'].value_counts().plot(kind='pie', autopct='%1.
1f%%')
plt.title('Distribution of Brands')
plt.ylabel('')
plt.show()
```



3. **Normalization of Numerical Features:**
   The 'Price' and 'Rating' columns were normalized using MinMaxScaler to ensure all features contribute equally to the model training.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
columns_to_normalize = ['Price', 'Rating']
df[columns_to_normalize] = scaler.fit_transform(df[colum
ns_to_normalize])
```

4. **Dataset Preparation for Surprise Library:**
   The normalized ratings were then used to create a dataset suitable for collaborative filtering using the Surprise library.

   ```
   from surprise import Dataset, Reader
   reader = Reader(rating_scale=(0, 5))
   data = Dataset.load_from_df(df[['User ID', 'Brand', 'Rating']], reader)
   ```

## 2.2 Model Training

1. **Train-Test Split:**
   The dataset was divided into training and testing subsets, with 80% of the data used for training and 20% for testing.

   ```
   from surprise.model_selection import train_test_split
   trainset, testset = train_test_split(data, test_size=0.2)
   ```

2. **Collaborative Filtering Algorithm:**
   The SVD (Singular Value Decomposition) algorithm was selected for building the recommendation system.

   ```
   from surprise import SVD
   algo_cf = SVD()
   algo_cf.fit(trainset)
   ```

## 2.3 Recommendation Function

A function was implemented to generate brand recommendations for a specific user. It utilized collaborative filtering predictions to filter and rank recommendations based on user preferences.

```
def brand_recommendation(user_id, num_recommendations):
    cf_predictions = algo_cf.test(testset)
    # Filter and sort predictions...
```

## 2.4 Model Evaluation

1. **Root Mean Square Error (RMSE) and Mean Absolute Error (MAE):**
   The RMSE and MAE were calculated to assess the predictive accuracy of the model.

   ```
   from surprise import accuracy
   accuracy.rmse(predictions)
   accuracy.mae(predictions)
   ```

---

## 3. Results

- **Data Summary:**
  The dataset comprised various products with their respective user ratings. Key insights were visualized to understand user preferences across different brands and categories.
- **Recommendations:**
  Upon testing the brand recommendation function for a user (e.g., User ID 7), the top recommendations along with the estimated ratings were successfully generated.
- **Evaluation Metrics:**
  - **RMSE:** 0.31099155500406617
  - **MAE:** 0.26456351782828913