Raúl Martin, Noel Pedrosa i Adrià Porta

README for IRWA Project - Part 1: Data Preparation and Exploratory Data Analysis (EDA)

Introduction

This README explains how to run the code in this notebook as well as how to select the different functions, algorithms or options that change both the ranking scores and exploratory analysis.

Step 1: Set Up Environment

Before running the code, you have to install the required libraries if you have not installed them yet. You can use Google Colab or a Jupyter notebook.

```python
pip install pandas nltk matplotlib seaborn wordcloud
```

These libraries are required for:

- **pandas**: Data manipulation
- **nltk**: Text processing (stopwords, tokenization, stemming)
- **matplotlib**: Plotting and visualizations
- **seaborn**: Statistical data visualization
- **wordcloud**: Generating word clouds

Once installed, import the necessary libraries in your notebook:

```python
import pandas as pd
import numpy as np
import nltk
import string
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from wordcloud import WordCloud
import json
```

Step 2: Load the Dataset

1. **Upload the dataset**: The first thing you need to do is upload the given dataset `fashion_products_dataset.json`. This dataset can be uploaded directly via Google Colab's file upload dialog.

2. **Loading the dataset in your notebook**:

```python
from google.colab import files
import json

uploaded = files.upload()  # Opens a file-picker dialog in
Colab
file_name = list(uploaded.keys())[0]
with open(file_name, 'r') as f:
    data = json.load(f)

df = pd.DataFrame(data)
```

Step 3: Preprocess Text Fields (Title and Description)

You will need to preprocess the text fields (`title`, `description`). For that there is the `clean_text()` function does the following:

- Removes stop words
- Tokenizes the text
- Removes punctuation
- Applies stemming

You can run the following code to preprocess the data:

```python
stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def clean_text(text):
    if not isinstance(text, str):
        return ""
    text = text.lower()
    text = text.translate(str.maketrans('', '',
string.punctuation))
    tokens = nltk.word_tokenize(text)
    tokens = [stemmer.stem(word) for word in tokens if word
not in stop_words]
    return " ".join(tokens)

df['clean_title'] = df['title'].apply(clean_text)
df['clean_description'] = df['description'].apply(clean_text)
```

Step 4: Exploratory Data Analysis (EDA)

Top 10 Brands

For visualizing the top 10 brands in the dataset. This will give you an idea of the dominant brands in the product catalog.

```python
top_brands = df['brand'].value_counts().head(10)
sns.barplot(y=top_brands.index, x=top_brands.values)
plt.title("Top 10 Brands")
plt.show()
```

Rating Distribution

Visualize how the ratings are distributed for the products in the dataset:

```python
df['average_rating'] = pd.to_numeric(df['average_rating'],
errors='coerce')
sns.histplot(df['average_rating'], bins=10)
plt.title("Average Rating Distribution")
plt.show()
```

Price and Discount Analysis

Visualize the distribution of the product prices and discounts.

```python
df['selling_price'] = df['selling_price'].str.replace(',',
'').astype(float)
sns.boxplot(x=df['selling_price'])
plt.title("Selling Price Distribution")
plt.show()
```

Word Cloud

Generate a word cloud from the product titles to see the most common words.

```python
all_titles = " ".join(df['clean_title'])
WordCloud(width=800, height=400,
background_color='white').generate(all_titles).to_image()
```

Step 5: Saving the Cleaned Dataset

After preprocessing and visualizing, there is the code to save the cleaned dataset to a CSV for further use in the later parts of the project.

```python
df.to_csv("fashion_products_clean.csv", index=False)
```