import pandas as pd

import numpy as np

import matplotlib as mpl

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

from textblob import TextBlob

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import linear\_kernel

# Calling Clean Data

data = pd.read\_csv('cleaned\_data.csv')

data.shape

data.head()

data.describe()

data.describe().T.drop('count',axis=1)

data['Clothing ID'].unique()

data['Clothing ID'].value\_counts()

data = data[['Clothing ID', 'Title', 'Review Text','Rating']]

# Getting Feedback of customers about Clothing ID

data = data.dropna(subset=['Review Text'])

# TF-IDF used for feacture Vectorizer scikit-learn

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(data['Review Text'])

#Applying Cosine Similarity

cosine\_similarities = linear\_kernel(tfidf\_matrix, tfidf\_matrix)

def recommend\_items(Clothing\_id, cosine\_similarities, data, top\_n=5):

indices = pd.Series(data.index, index=data['Clothing ID']).drop\_duplicates()

if Clothing\_id not in indices:

print(f"Clothing ID {Clothing\_id} not found in the data.")

return None

idx = indices[Clothing\_id]

# Locating the cosine similarities for the specific Clothing\_id

similarity\_scores = cosine\_similarities[idx, :]

# Displaying the indices of the top\_n items with highest similarity

top\_items\_indices = similarity\_scores.argsort()[::-1][1:top\_n+1].flatten()

return data.iloc[top\_items\_indices]

Clothing\_id = 420 # Selected Clothing ID

recommended\_items = recommend\_items(Clothing\_id, cosine\_similarities, data)

if recommended\_items is not None:

print(recommended\_items)

# Visualising the recommended Clothing ID

plt.figure(figsize=(12, 6))

plt.bar(recommended\_items['Clothing ID'], recommended\_items['Rating'], color='red')

plt.xlabel('Clothing ID')

plt.ylabel('Rating')

plt.title(f'Recommended Items for Clothing ID {Clothing\_id}')

plt.xticks(rotation=45)

plt.grid(axis='y')

plt.show()