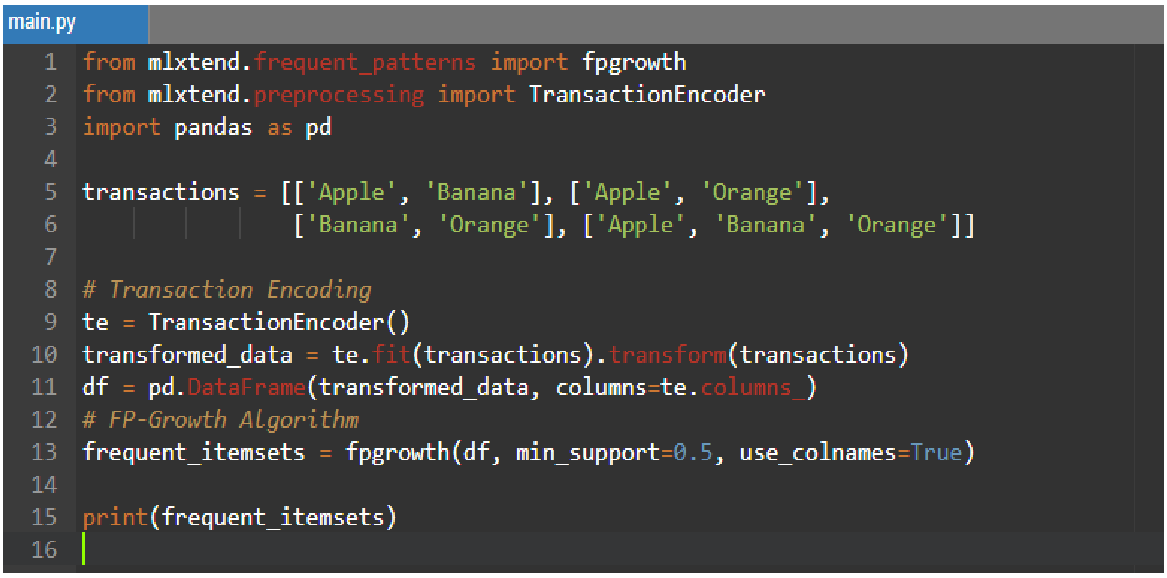
1. You are given a dataset of movies with various attributes like genres, keywords, and descriptions. Your task is to build a content-based recommendation engine that recommends movies similar to a given movie based on these attributes.

Steps:

- Preprocess the Data: Extract relevant features (e.g., genres, overview).
- Vectorize the Text Data: Use TF-IDF on the `overview` field.
- Compute Similarity: Use cosine similarity to find similar movies.
- Recommend: Given a movie, recommend the top 10 most similar movies based on content.

Note: Use IMDB dataset

2. What will be the output of following code?



```python
from mlxtend.frequent_patterns import fpgrowth
from mlxtend.preprocessing import TransactionEncoder
import pandas as pd

transactions = [['Apple', 'Banana'], ['Apple', 'Orange'],
                ['Banana', 'Orange'], ['Apple', 'Banana', 'Orange']]

# Transaction Encoding
te = TransactionEncoder()
transformed_data = te.fit(transactions).transform(transactions)
df = pd.DataFrame(transformed_data, columns=te.columns_)
# FP-Growth Algorithm
frequent_itemsets = fpgrowth(df, min_support=0.5, use_colnames=True)

print(frequent_itemsets)

```

3. Build an item-based collaborative filtering recommendation engine. Instead of recommending items based on similar users, recommend items that are similar to those that a user has already interacted with.

Steps:

- Preprocess the Data: Create a user-item matrix where rows are users and columns are items (movies).
- Compute Item Similarity: Calculate similarity between items based on user interactions.
- Recommend Items: For a given user, recommend items that are similar to those the user has already rated highly.

4. Using the mlxtend library, write a Python program to generate association rules from a dataset of transactions. The program should allow setting a minimum support threshold and minimum confidence threshold for rule generation.

transactions = [['Tea', 'Bun'], ['Tea', 'Bread'], ['Bread', 'Bun'], ['Tea', 'Bun', 'Bread']]

5. Build a popularity-based recommendation system. The system should recommend movies based on their overall popularity (e.g., number of ratings or average rating).

Steps:

- Preprocess the Data: Calculate the total number of ratings and average rating for each movie.
- Rank the Movies: Rank movies based on the chosen popularity metric.
- Recommend Movies: Recommend the top N most popular movies to any user.