Introduction:

Collaborative filtering technique is a part of recommender systems that can be used for recommending similar items when users enrolls to an item and suggesting items to registered user who has some prior enrollments. Collaborative filtering technique helps in finding out the item-item similarity as well as user-user similarity from the given dataset. For implementing Collaborative filtering out of various available libraries **benfred/implict** library has been chosen as the preferred option.

GitHub Url:
https://github.com/benfred/implicit

Advantages:

benfred/implict has following advantages over other libraries:

1) The major advantage of implicit over other libraries is it can be applied for implicit feedback dataset(data gathered from user behaviour) as well as explicit dataset(data user directly inputs eg: ratings).

2) Better performance,multiple algorithms support and has decent amount of users.

Implementation:

Implicit implements the following popular recommendation algorithms for implicit feedback datasets:

- Alternating Least Squares .
- Bayesian Personalized Ranking.
- Item-Item Nearest Neighbour models using Cosine, TFIDF or BM25 as a distance metric.

Out of the available models **Alternating Least square** model is producing more accurate results with better performance for the sample data so its was chosen as preferred model. After initialising the model following actions needs to be performed.

1) Fit the input data into model

    Model.fit(item_users, show_progress=True)

2) Now for suggesting similar items

    Model.similar_items(itemid, N=10)

    Returns: List of (itemid, score) tuples

3) For recommending items to a user

    Model.recommend(userid, user_items, N=10, filter_already_liked_items=True, filter_items=None, recalculate_user=False).

    Returns: List of (itemid, score) tuples.

Sample POC Input and output:

1) for similar items:

Input:

Enter course id
13760(F406w- EasyLineproduct range)

Output:

1. F377w - EN - Enclosed switches applications.
2. F924w - EN - OS Switch fuses - Sales arguments.
3. F400w - EN - SlimLine XR - General Features.

2) For recommending items to user.

Input:

Enter User id
4

Output:

1. ABB in the Solar Inverter Space.
2. PC Power Enclosures.
3. Introduction to EP Building Products.

Python code:

```
import pandas as pd
from scipy.sparse import coo_matrix, csr_matrix
import implicit
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider
from implicit.nearest_neighbours import (BM25Recommender, CosineRecommender,
                                         TFIDFRecommender, bm25_weight)
from implicit.datasets.lastfm import get_lastfm



auth_provider = PlainTextAuthProvider(username='cassandra', password='cassandra')
cluster = Cluster(['107.170.83.67'], auth_provider=auth_provider)
session = cluster.connect()
session.set_keyspace('ie_transact_e1')
cluster.connect()
raw_data = pd.read_csv('abb_data_3.csv')
plays = raw_data[['user_id', 'course_id', 'play']].copy()
playsMatrix = coo_matrix((plays['play'], (plays['course_id'], plays['user_id'])))
playsMatrix = (bm25_weight(playsMatrix, B=0.9) * 5).tocsr()
model = implicit.als.AlternatingLeastSquares(factors=50)
data = model.fit(playsMatrix)
input_data = input("Please select an option \n 1.Item-Item recommendation \n
2.User-Item recommendation\n 3.Exit \n")
def get_ids(ars):
  arr = []
  for ar in ars:
      arr.append(ar[0])
  return arr
def display_results(arr):
    placeholders = ', '.join(['%s'] * len(arr))
    query = 'SELECT course_id,course_title FROM Enrollment_E1 WHERE SYS_KEY=1 AND
course_id IN ({}) GROUP BY course_id ALLOW FILTERING'.format(
        placeholders)
    results = session.execute(query, arr)
    lis = {}
    for row in results:
        lis[int(row[0])] = row[1]
    for ind,ar in enumerate(arr):
        if (ind > 0):
          print("%s. %s" % (ind,lis[ar]))
input_data = int(input_data)
while input_data == 1 or input_data == 2:
    if (input_data == 1):
        course_id = input("Enter course id\n")
        recommendations = model.similar_items(int(course_id),N=11)
        ids_list = get_ids(recommendations)
        display_results(ids_list)
        input_data = int(input(
            "Please select an option \n 1.Item-Item recommendation \n 2.User-Item
recommendation\n 3.Exit \n"))

    else:
        user_id = input("Enter User id\n")
        recommendations = model.recommend(int(user_id), playsMatrix, 10,
filter_already_liked_items=True)
        ids_list = get_ids(recommendations)
        display_results(ids_list)
        input_data = int(input(
            "Please select an option \n 1.Item-Item recommendation \n 2.User-Item
recommendation\n 3.Exit \n"))
```