Assignment No 5

```
import numpy as np
import pandas as pd
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics import mean_squared_error

data = {
   'user_id': [1, 1, 2, 2, 3, 3, 4],
   'item_id': [1, 2, 2, 3, 1, 3, 2],
   'rating':  [5, 4, 4, 5, 2, 4, 3]
}

df = pd.DataFrame(data)

user_item_matrix = df.pivot(index='user_id', columns='item_id', values='rating').fillna(0)
print("User-Item Matrix:\n", user_item_matrix, "\n")

svd = TruncatedSVD(n_components=2, random_state=42)
user_features = svd.fit_transform(user_item_matrix)

predicted_ratings = np.dot(user_features, svd.components_)
pred_df = pd.DataFrame(predicted_ratings,
                       index=user_item_matrix.index,
                       columns=user_item_matrix.columns)
print("Predicted Rating Matrix:\n", pred_df.round(2), "\n")

rmse = np.sqrt(mean_squared_error(user_item_matrix.values.flatten(),
                                  pred_df.values.flatten()))
print("RMSE:", round(rmse, 3), "\n")

def recommend_items(user_id, n=2):
   user_ratings = pred_df.loc[user_id]
   rated_items = df[df['user_id'] == user_id]['item_id'].values
   recommendations =
user_ratings.drop(rated_items).sort_values(ascending=False).head(n)
   return recommendations

print("Top Recommendations for User 1:")
print(recommend_items(1))
```

Viva-Ready Q&A

Q: What is collaborative filtering?
A: It recommends items based on other users' preferences — assuming users with similar tastes will like similar items.

Q: Why use SVD in recommendation systems?
A: SVD reduces dimensionality and finds hidden patterns between users and items, improving recommendation accuracy.

Q: What does `TruncatedSVD` do?
A: It performs low-rank approximation on large sparse matrices efficiently without computing the full decomposition.

Q: What are latent features in SVD?
A: They represent hidden factors — for example, users' preference strength for specific movie types or item categories.

Q: Why are missing ratings filled with 0 before SVD?
A: Because SVD requires a complete matrix. 0 indicates "no rating" rather than an actual preference.

Q: What is the meaning of RMSE here?
A: RMSE shows how close the reconstructed ratings are to the actual ratings — lower RMSE means better prediction accuracy.

Q: What does `n_components` represent?
A: The number of latent factors extracted by SVD — controlling how much of the variance is captured.

Q: What is the output of this system?
A: A list of top-N recommended items for each user, based on predicted ratings.

Q: Why is SVD better than simple similarity-based CF?
A: It captures global user–item relationships instead of only local (neighbor-based) ones, leading to more accurate predictions.

Q: What are limitations of this method?
A:

- **Needs enough rating data.**
- **Doesn't handle new users/items well (cold start).**
- **Assumes linear relationships.**

Q: How can the model be improved?
A:

- **Use more components (`n_components`).**
- **Try regularized matrix factorization (e.g., ALS, NMF).**
- **Incorporate implicit feedback (clicks, views).**