```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re

# Load datasets
ratings = pd.read_csv('/content/zee-ratings.dat', sep='::', engine='python',
                      names=['UserID', 'MovieID', 'Rating', 'Timestamp'])
users = pd.read_csv('/content/zee-users.dat', sep='::', engine='python',
                    names=['UserID', 'Gender', 'Age', 'Occupation', 'Zip-code'])
movies = pd.read_csv('/content/zee-movies.dat', sep='::', engine='python',
                     names=['MovieID', 'Title', 'Genres'], encoding='ISO-8859-1')


# Merge all data
data = ratings.merge(users, on='UserID').merge(movies, on='MovieID')
data['Rating'] = data['Rating'].astype(int)
data['Year'] = data['Title'].str.extract(r'\((\d{4})\)', expand=False).astype('Int64')



# Age Distribution
age_map = {1:"<18", 18:"18-24", 25:"25-34", 35:"35-44", 45:"45-49", 50:"50-55", 56:"56+"}
data['AgeGroup'] = data['Age'].map(age_map)

# Ratings per age group
age_rating_counts = data.groupby('AgeGroup')['Rating'].count().sort_values(ascending=False)
occupation_rating_counts = data.groupby('Occupation')['Rating'].count().sort_values(ascending=False)

# Top decades
data['Decade'] = (data['Year'] // 10) * 10

# Popular Movies
movie_stats = data.groupby('Title').agg({'Rating': ['count', 'mean']})
movie_stats.columns = ['RatingCount', 'AvgRating']
top_movie = movie_stats['RatingCount'].idxmax()



# Create pivot table
pivot_table = data.pivot_table(index='UserID', columns='Title', values='Rating')
target_movie = 'Liar Liar (1997)'
similar_movies = pivot_table.corrwith(pivot_table[target_movie])
similar_df = pd.DataFrame(similar_movies, columns=['PearsonCorr']).dropna()
similar_df = similar_df.sort_values('PearsonCorr', ascending=False).head(6)
```

```
/usr/local/lib/python3.11/dist-packages/numpy/lib/_function_base_impl.py:2914: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
/usr/local/lib/python3.11/dist-packages/numpy/lib/_function_base_impl.py:2773: RuntimeWarning: divide by zero encountered in divide
  c *= np.true_divide(1, fact)
/usr/local/lib/python3.11/dist-packages/numpy/lib/_function_base_impl.py:2773: RuntimeWarning: invalid value encountered in multiply
  c *= np.true_divide(1, fact)
/usr/local/lib/python3.11/dist-packages/numpy/lib/_function_base_impl.py:2922: RuntimeWarning: invalid value encountered in divide
  c /= stddev[:, None]
/usr/local/lib/python3.11/dist-packages/numpy/lib/_function_base_impl.py:2923: RuntimeWarning: invalid value encountered in divide
  c /= stddev[None, :]
```

```
from sklearn.neighbors import NearestNeighbors
from sklearn.metrics.pairwise import cosine_similarity
from scipy.sparse import csr_matrix

movie_user_matrix = pivot_table.fillna(0).T
csr_data = csr_matrix(movie_user_matrix.values)
knn = NearestNeighbors(metric='cosine', algorithm='brute')
knn.fit(csr_data)

# Recommend similar movies
def get_movie_recommendations(movie_title):
    movie_idx = movie_user_matrix.index.tolist().index(movie_title)
    distances, indices = knn.kneighbors(movie_user_matrix.iloc[movie_idx, :].values.reshape(1, -1), n_neighbors=6)
    return movie_user_matrix.index[indices.flatten()[1:]]

get_movie_recommendations("Liar Liar (1997)")
```

```
Index(['Mrs. Doubtfire (1993)', 'Ace Ventura: Pet Detective (1994)',
       'Dumb & Dumber (1994)', 'Home Alone (1990)', 'Wayne's World (1992)'],
      dtype='object', name='Title')
```

```
!pip install --force-reinstall --no-cache-dir numpy scikit-surprise
```

```
Collecting numpy
  Downloading numpy-2.3.1-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.1/62.1 kB 2.4 MB/s eta 0:00:00
Collecting scikit-surprise
  Downloading scikit_surprise-1.1.4.tar.gz (154 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 154.4/154.4 kB 9.1 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting joblib>=1.2.0 (from scikit-surprise)
  Downloading joblib-1.5.1-py3-none-any.whl.metadata (5.6 kB)
Collecting scipy>=1.6.0 (from scikit-surprise)
  Downloading scipy-1.16.0-cp311-cp311-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.9/61.9 kB 119.4 MB/s eta 0:00:00
Downloading numpy-2.3.1-cp311-cp311-manylinux_2_28_x86_64.whl (16.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.9/16.9 MB 232.1 MB/s eta 0:00:00
Downloading joblib-1.5.1-py3-none-any.whl (307 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.7/307.7 kB 310.2 MB/s eta 0:00:00
Downloading scipy-1.16.0-cp311-cp311-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (35.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 35.3/35.3 MB 219.1 MB/s eta 0:00:00
Building wheels for collected packages: scikit-surprise
  Building wheel for scikit-surprise (pyproject.toml) ... done
  Created wheel for scikit-surprise: filename=scikit_surprise-1.1.4-cp311-cp311-linux_x86_64.whl size=2469551 sha256=e78d350f2f3aac12d65
  Stored in directory: /tmp/pip-ephem-wheel-cache-e2jarp47/wheels/2a/8f/6e/7e2899163e2d85d8266daab4aa1cdabec7a6c56f83c015b5af
Successfully built scikit-surprise
Installing collected packages: numpy, joblib, scipy, scikit-surprise
  Attempting uninstall: numpy
    Found existing installation: numpy 2.0.2
    Uninstalling numpy-2.0.2:
      Successfully uninstalled numpy-2.0.2
  Attempting uninstall: joblib
    Found existing installation: joblib 1.5.1
    Uninstalling joblib-1.5.1:
      Successfully uninstalled joblib-1.5.1
  Attempting uninstall: scipy
    Found existing installation: scipy 1.15.3
    Uninstalling scipy-1.15.3:
      Successfully uninstalled scipy-1.15.3
  Attempting uninstall: scikit-surprise
    Found existing installation: scikit-surprise 1.1.4
    Uninstalling scikit-surprise-1.1.4:
      Successfully uninstalled scikit-surprise-1.1.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
cupy-cuda12x 13.3.0 requires numpy<2.3,>=1.22, but you have numpy 2.3.1 which is incompatible.
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.3.1 which is incompatible.
tensorflow 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 2.3.1 which is incompatible.
Successfully installed joblib-1.5.1 numpy-2.3.1 scikit-surprise-1.1.4 scipy-1.16.0
WARNING: The following packages were previously imported in this runtime:
  [joblib,numpy,scipy]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
!pip install lightfm
```

```
Collecting lightfm
  Downloading lightfm-1.17.tar.gz (316 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 316.4/316.4 kB 5.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from lightfm) (2.3.1)
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.11/dist-packages (from lightfm) (1.16.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from lightfm) (2.32.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (from lightfm) (1.6.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->lightfm) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->lightfm) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->lightfm) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->lightfm) (2025.6.15)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn->lightfm) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn->lightfm) (3.6.0)
Building wheels for collected packages: lightfm
  Building wheel for lightfm (setup.py) ... done
  Created wheel for lightfm: filename=lightfm-1.17-cp311-cp311-linux_x86_64.whl size=831128 sha256=8fe3936cb50626a7d0a0206c4de834c66aa2d
  Stored in directory: /root/.cache/pip/wheels/b9/0d/8a/0729d2e6e3ca2a898ba55201f905da7db3f838a33df5b3fcdd
Successfully built lightfm
Installing collected packages: lightfm
Successfully installed lightfm-1.17
```

```
import pandas as pd

ratings = pd.read_csv('/content/zee-ratings.dat', sep='::', engine='python',
                      names=['UserID', 'MovieID', 'Rating', 'Timestamp'],
                      encoding='ISO-8859-1')
```

```
# Check unique values in the Rating column
print(ratings['Rating'].unique()[:10])  # just to inspect

# Drop rows where 'Rating' column contains non-numeric entries (like 'Rating')
ratings = ratings[ratings['Rating'] != 'Rating']

# Convert to float
ratings['Rating'] = ratings['Rating'].astype(float)

 #Setup dataset
dataset = Dataset()
dataset.fit(ratings['UserID'], ratings['MovieID'])

# Build interactions
(interactions, weights) = dataset.build_interactions(
    [(x.UserID, x.MovieID, x.Rating) for x in ratings.itertuples()]
)

# Train model
model = LightFM(no_components=4, loss='warp')
model.fit(interactions, epochs=10, num_threads=2)
```

```
[5. 3. 4. 2. 1.]
<lightfm.lightfm.LightFM at 0x7b3b063e3cd0>
```

```
import pandas as pd

# Load data
ratings = pd.read_csv('/content/zee-ratings.dat', sep='::', engine='python',
                      names=['UserID', 'MovieID', 'Rating', 'Timestamp'],
                      encoding='ISO-8859-1', header=None)

# Drop erroneous rows
ratings = ratings[ratings['Rating'] != 'Rating']

# Convert to correct datatype
ratings['UserID'] = ratings['UserID'].astype(int)
ratings['MovieID'] = ratings['MovieID'].astype(int)
ratings['Rating'] = ratings['Rating'].astype(float)
```

```
/tmp/ipython-input-10-4041570417.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  ratings['UserID'] = ratings['UserID'].astype(int)
```

```
from lightfm import LightFM
from lightfm.data import Dataset

# Step 1: Prepare dataset
dataset = Dataset()
dataset.fit(ratings['UserID'], ratings['MovieID'])

# Step 2: Create interaction matrix
```

```
(interactions, weights) = dataset.build_interactions(
    [(x.UserID, x.MovieID, x.Rating) for x in ratings.itertuples()]
)

# Step 3: Train MF model (WARP loss optimizes ranking)
model = LightFM(no_components=4, loss='warp')
model.fit(interactions, epochs=10, num_threads=2)
```

```
<lightfm.lightfm.LightFM at 0x7b3b077cbe50>
```

```
import numpy as np

# Get reverse mappings
user_id_map, user_feature_map, item_id_map, item_feature_map = dataset.mapping()
inv_item_map = {v: k for k, v in item_id_map.items()}

def recommend_movies(user_id, model, interactions, dataset, n=5):
    n_users, n_items = interactions.shape
    user_index = user_id_map[user_id]  # map to internal index

    # Predict scores for all items
    scores = model.predict(user_index, np.arange(n_items))
    top_items = np.argsort(-scores)[:n]

    return [inv_item_map[i] for i in top_items]



# Load movies
movies = pd.read_csv('/content/zee-movies.dat', sep='::', engine='python',
                     names=['MovieID', 'Title', 'Genres'], encoding='ISO-8859-1')

# Example: Recommend for user 42
user_recs = recommend_movies(42, model, interactions, dataset)

# Show titles
movies[movies['MovieID'].isin(user_recs)][['MovieID', 'Title']]
```

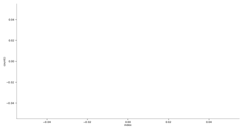
No entries Filter

index	MovieID	Title

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Time series



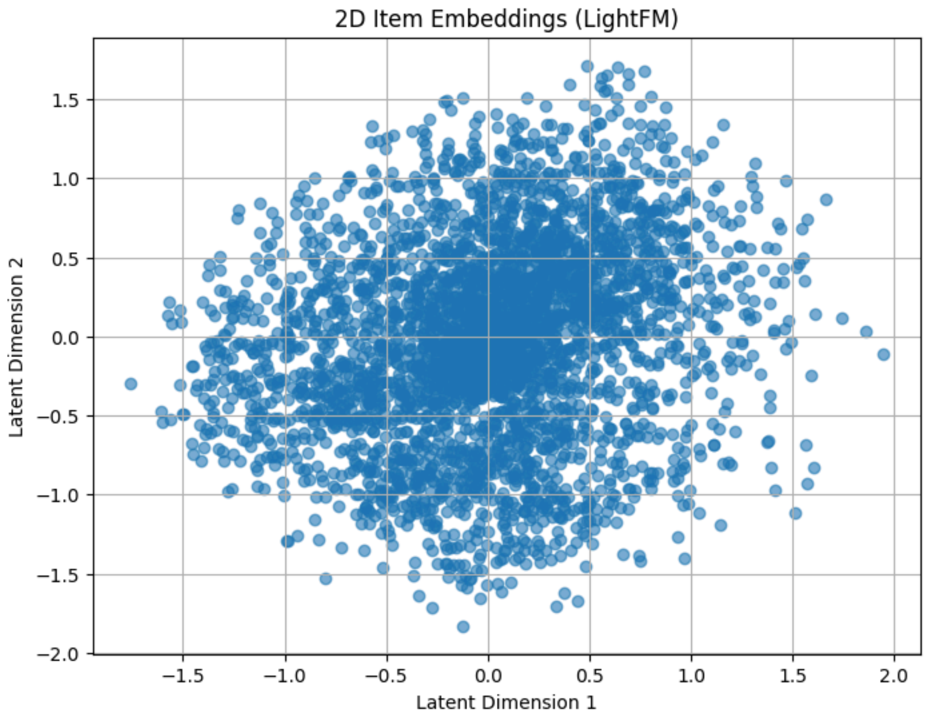
```
import matplotlib.pyplot as plt

user_embeddings = model.pu  # (n_users x d)
movie_embeddings = model.qi  # (n_movies x d)

# 2D visualization
plt.scatter(movie_embeddings[:, 0], movie_embeddings[:, 1], alpha=0.5)
plt.title('Movie Embeddings (d=2)')
plt.xlabel('Latent Feature 1')
plt.ylabel('Latent Feature 2')
plt.grid()
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipython-input-14-3200370509.py in <cell line: 0>()
      1 import matplotlib.pyplot as plt
      2 
----> 3 user_embeddings = model.pu  # (n_users x d)
      4 movie_embeddings = model.qi  # (n_movies x d)
      5 

AttributeError: 'LightFM' object has no attribute 'pu'
```

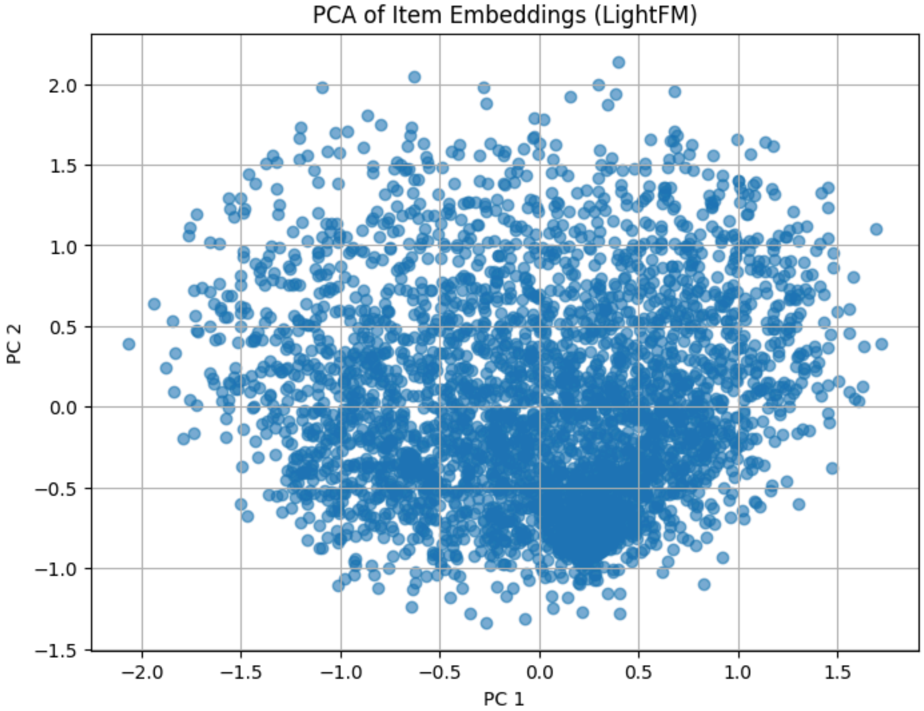
Next steps: Explain error

```
# Get user and item embeddings from LightFM
user_embeddings = model.user_embeddings  # shape: (num_users, no_components)
item_embeddings = model.item_embeddings  # shape: (num_items, no_components)
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.scatter(item_embeddings[:, 0], item_embeddings[:, 1], alpha=0.6)
plt.title("2D Item Embeddings (LightFM)")
plt.xlabel("Latent Dimension 1")
plt.ylabel("Latent Dimension 2")
plt.grid(True)
plt.show()
```



```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
item_emb_2d = pca.fit_transform(item_embeddings)

plt.figure(figsize=(8, 6))
plt.scatter(item_emb_2d[:, 0], item_emb_2d[:, 1], alpha=0.6)
plt.title("PCA of Item Embeddings (LightFM)")
plt.xlabel("PC 1")
plt.ylabel("PC 2")
plt.grid(True)
plt.show()
```


PCA of Item Embeddings (LightFM)
PC 2
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
PC 1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5