



MITx 6.86x

**Machine Learning with Python-From Linear Models to Deep Learning**[Course](#)[Progress](#)[Dates](#)[Discussion](#)[Resources](#)[Course](#) / [Unit 4. Unsupervised Learning...](#) / [Project 4: Collaborative Filtering vi](#)[< Previous](#)

## 1. Introduction

[Bookmark this page](#)

Your task is to build a mixture model for collaborative filtering. You are given a data matrix of ratings made by users where the matrix is extracted from a much larger Netflix database. The database has rated only a small fraction of the movies so the data matrix is only partially filled. The task is to fill the remaining entries of the matrix.

You will use mixtures of Gaussians to solve this problem. The model assumes that each user is a sample from a mixture model. In other words, we have  $K$  possible types of users and, for each user, we must sample a user type and then the rating profile from the Gaussian distribution of that type. We will use the Expectation Maximization (EM) algorithm to estimate such a mixture model from the observed rating matrix. The EM algorithm proceeds by iteratively assigning (softly) users to the Gaussian type they most likely belong to (E-step) and subsequently re-estimating the Gaussians associated with each type (M-step). Once we have estimated the model, we can use it to predict values for all the missing entries in the data matrix.

### Setup:

As with the last project, please use Python's **NumPy** numerical library for handling arrays and use **matplotlib** for producing figures and plots.

1. *Note on software:* For all the projects, we will use python 3.6 augmented with the **NumPy** library and the **matplotlib** plotting toolbox. In this project, we will also use the `typing` library, which is not part of the standard library (no need to install anything).
2. Download [netflix.tar.gz](#) and untar it in to a working directory. The archive contains the

- `kmeans` where we have implemented a baseline using the K-means algorithm
- `naive_em.py` where you will implement a first version of the EM algorithm (tabs 3-4)
- `em.py` where you will build a mixture model for collaborative filtering (tabs 7-8)
- `common.py` where you will implement the common functions for all models (tab 5)
- `main.py` where you will write code to answer the questions for this project



**edX**

test.py where you will write code to test your implementation of EM for a given test dataset. Additionally, you are provided with the following data files:

[About](#)

[Affiliates](#)

[edX for Business](#)

[Open edX](#)

[Careers](#)

[News](#)

- `netflix_incomplete.txt` a 2D dataset that you will work with in tabs 2-5

- `netflix_incomplete.txt` the netflix dataset with missing entries to be completed

- `netflix_complete.txt` the netflix dataset with missing entries completed

- `test_incomplete.txt` a test dataset to test for you to test your code against our

# Connect

[< Previous](#)[Next >](#)[Blog](#)[Contact Us](#)[Help Center](#)[Security](#)[Media Kit](#)

© 2023 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)