# Case Study 4.1 - Movies


## Helpful Links

- MITx Case Study Page
- Discussion Forum
- Case Study Evaluation Rubric
- Frequently Asked Questions

## Table of Contents


- Introduction
  - Logistics
  - Setup
- During
  - Import
  - Data
  - Cross Validation
  - Model 1: Random
  - Model 2: User-Based Collaborative Filtering
  - Model 3: Item-Based Collaborative Filtering
  - Model 4: Matrix Factorization
  - Precision and Recall @ k
  - Top- n Predictions
- Conclusion
  - Submitting

- Next Steps
- FAQ
- Troubleshooting


# Introduction

The purpose of this document is to show you how to complete the first graded case study of the course. The goal of this case study is to **experiment with various recommendation models for predicting user preferences for movies.**

You will gain practical experience in:

- **loading** a large dataset
- **creating** various models
- **evaluating** the predictions of those models
- **utilizing** tools like cross validation to train and test models
- **understanding** metrics like RMSE, precision and recall `@k` and top `n` recommendations

## Logistics

Here is how this case study will work.

You can follow along with the instructions in this document. They correspond to the code provided and activity template you will fill out to complete the case study. We have blank sections in the code where you can write your responses to our questions. These sections will be marked **RED** so you won't miss them. Some responses will include code, while others will not. We do our best to make this clear where needed.

Once you complete the case study, you will submit your work to the online MITx platform. From there, the system will assign you 2 peers to review. You will grade their submissions according to the provided evaluation rubric. **We recommend you check out this rubric now, so you know how you will be graded.**

Here are the key dates for this case study:

1. **By 23:59 UTC, Sunday, June 7, 2020.** Submit your case study to MITx. Be sure to convert to your local time zone.

2. **By 23:59 UTC, Monday, June 8, 2020.** Complete the peer reviews for 2 of your peers. (If the system does not give you 2 reviews, see the ***Next Steps*** section below for what to do. But, it is critical you complete any reviews you are assigned.)

## Setup

Follow these steps to get set up with the code and submission template for the case study.

1. If you have not already done so, follow the instructions for getting started with Microsoft Azure Notebooks. You will need to create and confirm your account there before proceeding to step 2.

2. **Clone** the case study library, which can be found here. Go to that page and click the *Clone* button to copy the code to your personal library. The dialog should look something like this:


3. **Select your language.** We support **both** Python and R in this case study. The remainder of this document will provide instructions for both languages, in the following format:

| Python | R |
|---|---|
| Some python information here... | Some R information here... |

4. **Select your experience level.** We have support for both *beginner* and *advanced* users of Python and R in this case study. Use the following chart to see the name of the notebook you should use, based on your experience level.

| Proficiency | Python | R |
|---|---|---|

| Proficiency | Python | R |
|---|---|---|
| Beginner | `beginner_python.ipynb` | `beginner_r.ipynb` |
| Advanced | `advanced_python.ipynb` | `advanced_r.ipynb` |

5. Open the notebook file you selected from the table above. This will start up the Jupyter notebook. If you need a refresher on how to work with a notebook, we have information for you here.

6. As described above, this notebook file will be the template you use to record your answers. So, before you do anything else, be sure to fill in the information in the ***Identification Information*** section at the top of the notebook.

7. Next, go to the ***Setup*** section and follow the instructions. This will **install** all the packages you need to complete the remainder of the case study.

# During

Great! Now, you should be all set up and ready to complete this case study. As stated above, we are going to be using a recommendation system to predict movies for users, based on an existing dataset. This document will walk you through how to work with this data and framework to arrive at some meaningful results!

**Remember our tip from our first Microsoft Azure Notebooks tutorial: *save your notebook often* to avoid losing your work!**

## Import

One of the first steps in any data science task is importing the necessary tools you will use.

See the ***Import*** section of your notebook and follow the instructions to import the required tools. See the table below for these requirements, for your reference.

| Python | R |
|---|---|

| Python | R |
|---|---|
| surprise | recommenderlab |

## Data

See the ***Data*** section of your notebook and follow the instructions to load the MovieLens data. You can read more about this dataset here, if you are interested.

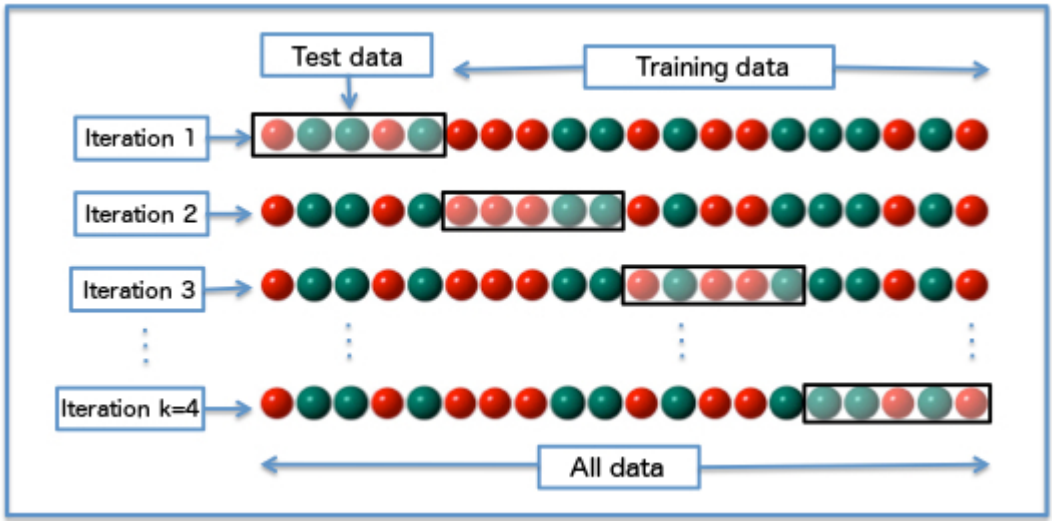
## Cross Validation

**Note: cross validation is currently only supported in the Python code. The R code uses a simple train-test split. If you are using R, you can skip ahead to the *Model 1: Random* section below.**

We will be using cross validation a lot in this code in the training and evaluation of our models. This strategy builds upon the idea of a **train-test** split, which you should already be familiar with.

Instead of doing 1 data split, though, we will do several of them. Each split of the data is called a ***fold***. We let `k` denote the number of folds we use. `k=5` is a common number to use.

This image provides a visual explanation of how cross validation works.




## Model 1: Random

We want to first get a baseline value for our model. What better way to do that than with a *random* algorithm! Essentially, this first algorithm is not personalized to the desires of any users - we just assign them movie ratings based on the initial distribution of the data.

See the ***Model 1: Random*** section of your notebook and follow the instructions to create a new model, train it on the data and evaluate the RMSE.

## Model 2: User-Based Collaborative Filtering

Surely, we can do much better than guessing the movie ratings randomly! Our next model will use the user-user defined notion of similarity to implement collaborative filtering.

See the ***Model 2: User-Based Collaborative Filtering*** section of your notebook and follow the instructions to create a new model, train it on the data and evaluate the RMSE.

## Model 3: Item-Based Collaborative Filtering

Our next model will use the item-item defined notion of similarity to once again implement collaborative filtering.

See the ***Model 3: Item-Based Collaborative Filtering*** section of your notebook and follow the instructions to create a new model, train it on the data and evaluate the RMSE.

## Model 4: Matrix Factorization

Our final model for this case study will use the matrix factorization approach with the SVD algorithm to try to predict user's movie ratings. Here, we try to determine some underlying mathematical structure in the user rating matrix, which can help us predict missing ratings in the future.

See the ***Model 4: Matrix Factorization*** section of your notebook and follow the instructions to create a new model, train it on the data and evaluate the RMSE.

## Precision and Recall @ k

RMSE is not the only metric we can use here. We can also examine two fundamental measures, precision and recall. We also add a parameter `k` which is helpful in understanding problems with multiple rating outputs.

See the ***Precision and Recall @ `k`*** section of your notebook and follow the instructions to compute various precision/recall values at various values of `k`.

### Top-`n` Predictions

Finally, we want to actually see what ratings the model predicts for our users. We can vary the amount of top movies we see per user by varying the value of `n`.

See the ***Top-`n` Predictions*** section of your notebook and follow the instructions to compute rating predictions for some users.

## Conclusion

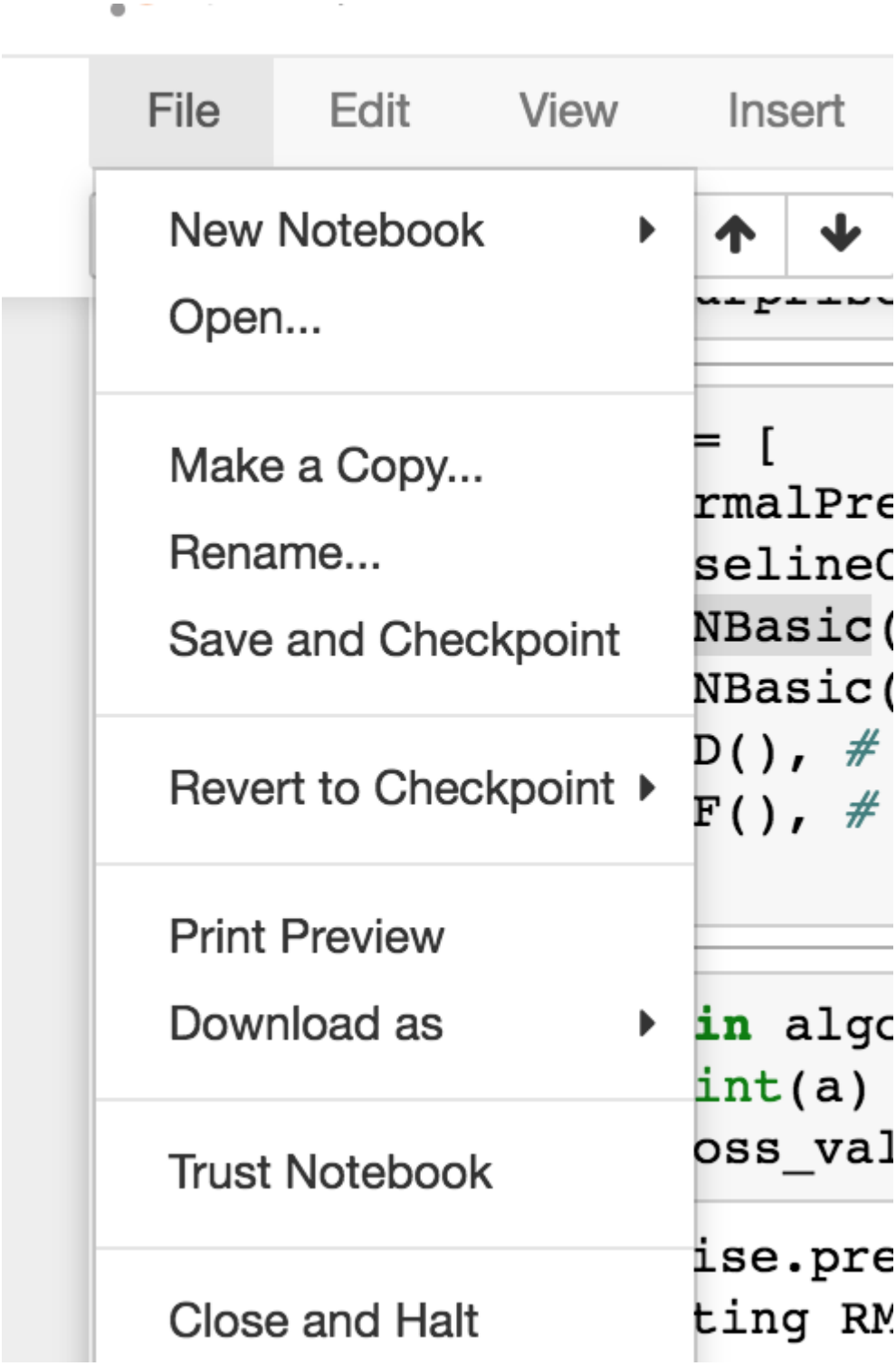
We really hope you enjoyed completing this case study and feel more comfortable about some of the learning objectives listed above.

There are just a few more things you need to know before you are done.

### Submitting

Follow these instructions to submit your responses to the MITx platform.

1. **Export your work.** In your notebook file on Microsoft Azure Notebooks, go to *File* > *Download As* then select **HTML**.




2. **Convert your file to a PDF.** You can use this tool to do this. The MITx platform doesn't allow HTML uploads, so we must take this in-between step.

3. **Name your file correctly.** We ask that you name your file in the following format:

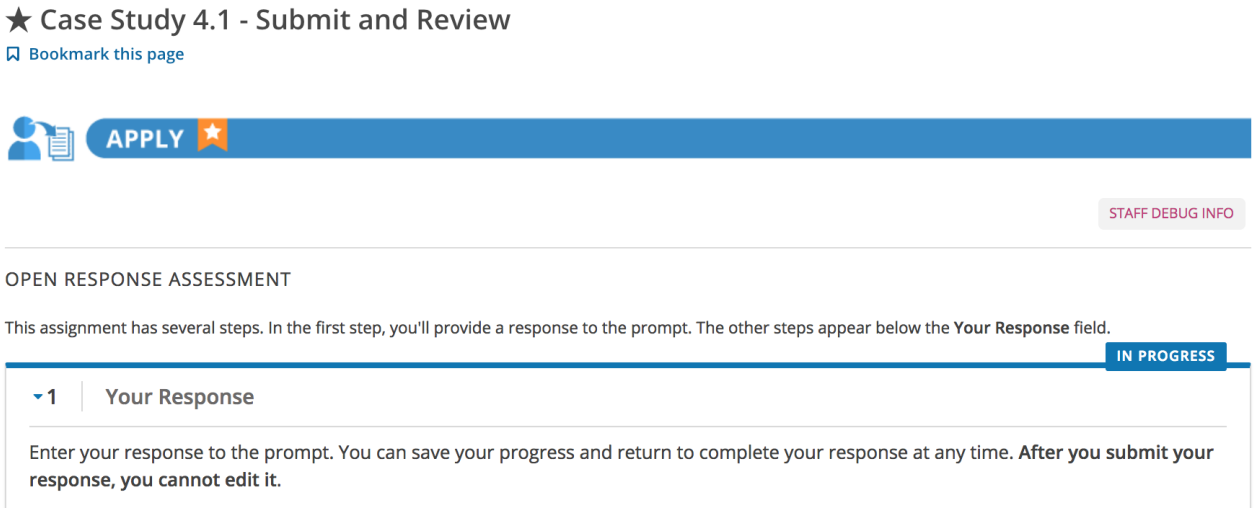
```
41_LastName_FirstName_mitxprousername.pdf
```

For example, if you are Bob Smith and your MITx **username** is bobs123, then you should save your file with this name:

```
41_Smith_Bob_bobs123.pdf
```

This makes it easier for other students **and** course TA's to identify your work and make sure you receive as much credit as possible. **You can find your MITx username on your progress page here, just to the left of your e-mail address.**

4. **Upload your file.** Go to the Case Study page (see Helpful Links above), then click the *right tab*. You should see a page that looks like this:




   - You can leave the *Your Response* field blank.
   - Click the *Choose Files* button and select the file from your computer. This is the file you named properly in step 2 above.
   - In the "Describe" text box, enter a description that exactly matches the **name of your file** (i.e. `41_Smith_Bob_bobs123.pdf` ).
   - Click the *Save your progress* button.
   - Then, click the *Upload Files* button. If you have any issues with this, fill out the *Issue Submission* form at the bottom of the page.
   - You can **re-upload as many times as you wish** before you submit. See the next bullet on how to submit.
   - Once you are confident with the last file you have uploaded, press the blue button that says *Submit your response and move on to the next step*. If you have any issues with this, fill out the *Issue Submission* form at the bottom of the page.

## Next Steps

Remember - after you submit your own work, be prepared for the system to assign you peers to review.

If the system does not give you any peers to review right away, we ask that you check back to the site again to see if you have any new reviews. It may take a few hours for a new review to come in.

## FAQ

Here are some questions we see from students often and our answers to those questions. This list is subject to change.

---

Q: **I get an error about the request being too long and the notebook won't load. What do I do?**

You should try to clear the cache and cookies in your browser. This page has a tutorial on how to do this.

---

Q: **I see an error like this when I try to install `surpriselib`. What do I do?**

```
grpcio 1.11.0 has requirement protobuf>=3.5.0.post1,
but you'll have protobuf 3.4.1 which is incompatible.
```

Don't worry! This is not an error, just a warning. (The red text is annoying, we know.) As long as you can keep running the cells below, after the installation completes, you should be all set.

---

Q: **I see a bunch of red text when I run a cell. It looks like an error. Does this mean my code is broken?**

A: No, this is normal! These are just warning messages from the code. As long as the cell runs and you receive a success message, you are good to go!

---

Q: **I can't get my code to work. I keep getting a package error or some other error I don't know how to solve. What do I do?**

A: See the troubleshooting section below for help.

---

Q: **The numbers I am getting on the case study vary slightly from the ones other people are getting. Is this normal?**

A: Yes, this is completely normal! This has to do with the way is randomly split, or the way the model trains based on the data. As long as your numbers are reasonably close to the values of others, you should be all set.

Also, please note that the R and Python values may vary in comparison to one another, due to the specific algorithms that each language uses. This is also completely acceptable, as long as you have code and results to support your results.

---

Q: **The histogram image isn't showing up in my R notebook when I generate the HTML output. What's going on?**

A: This is an issue with Microsoft Azure Notebooks, not you. Don't worry. We have added language in the evaluation rubric to still give you full credit for this.

---

Q: **It's getting really close to the submission deadline and I am struggling to make progress on the assignment. Can I get an extension? What should I do?**

A: Unfortunately, due to the nature of the course, it is our official policy that **no extensions will be granted** on graded assignments.

However, we encourage you to submit what you have so far. Then, if you are concerned about your grade, you can contact us privately with our G-Mail address.

## Troubleshooting

Having trouble getting your code to run? Questions about a particular part of the case study? Not able to submit your assignment to MITx? Your course TA's are here to help!

Please follow these steps if you get stuck.

1. Make sure you followed all of the instructions exactly as specified.
2. **Search** the Discussion Forum to see if your question has already been answered.
3. Please post in the **Module 4: Case Studies** section of the Discussion Forum and your (*new*, *unanswered*) question will be answered as soon as possible!

---