Movie Recommendations

Veeral Patel

Regis University

April 22nd, 2021

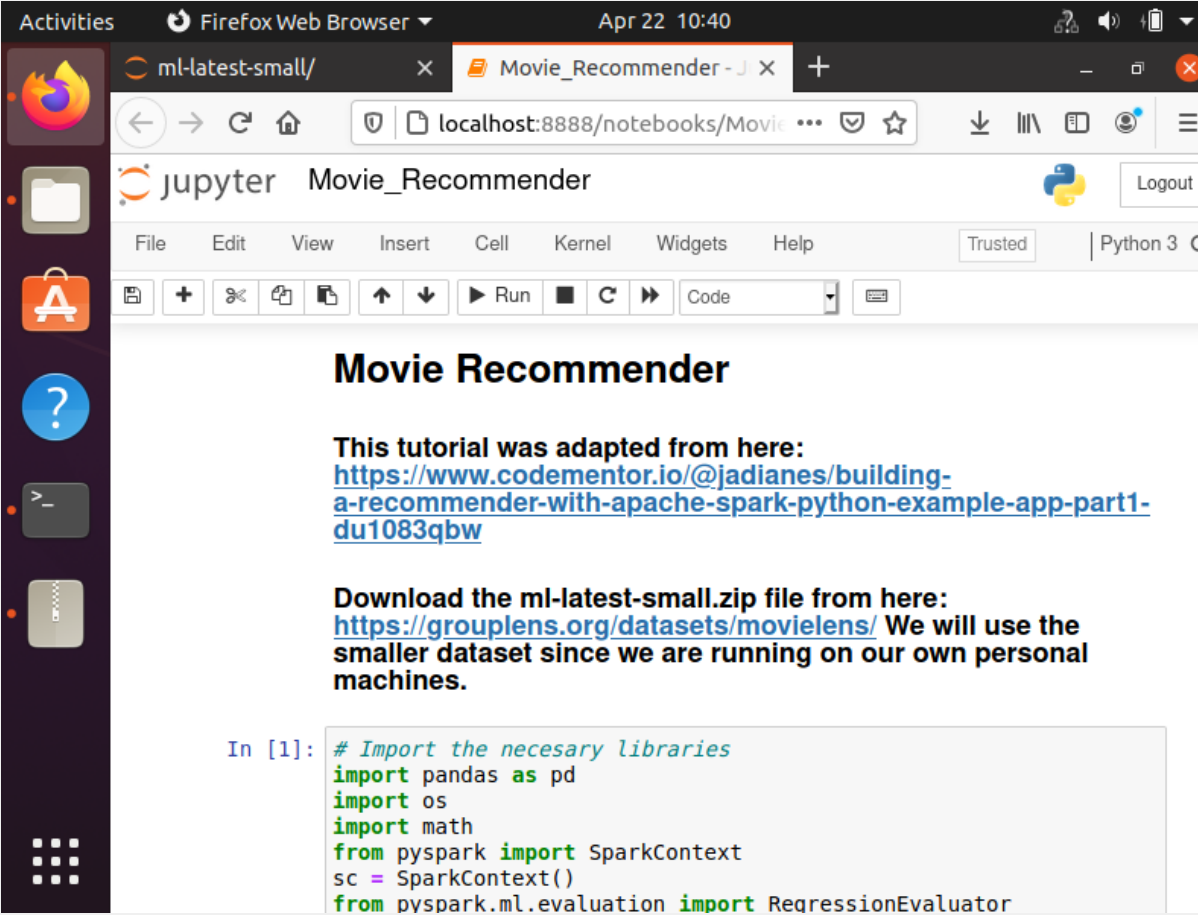
1. **Introduction**

For this project, we were to use a VM that we had installed a few weeks ago. This week, we learned how to build our own recommendation model. For this assignment, we downloaded a pre created folder that had a few.csv files in it. These files contained data for movies. In this homework, we took the moves as well as their ratings and then used userID, movieID, rating that the user gave that movie. This helped build a model with what type of moves the user ranks higher. Then the model uses this data in order to suggest moves for the specific user to watch in the future. In this project, we used ML libraries, pyspark, and

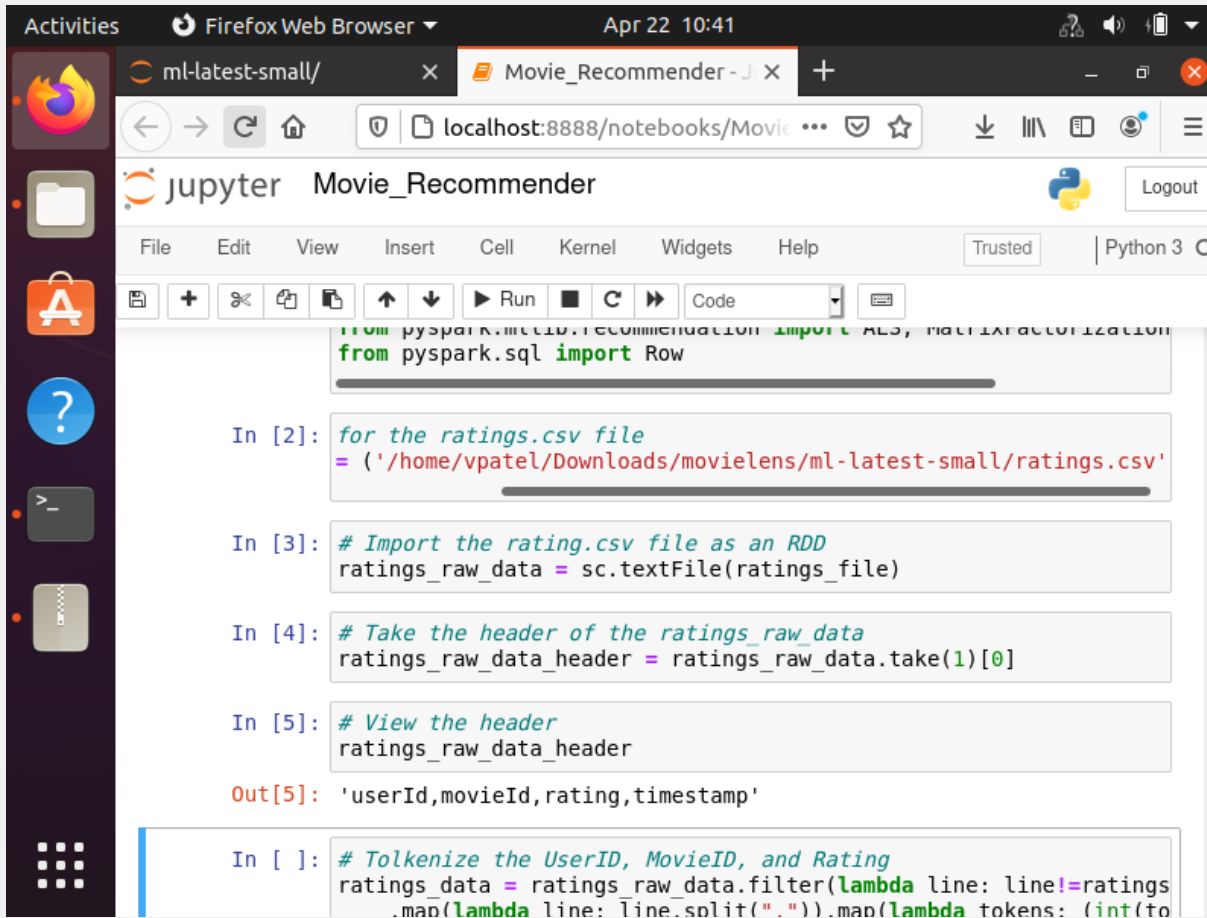
1. **Methods/Code/Screenshots**

I used a windows computer and I had to install pyspark. I used the PDF that was given to the class as well as some tutorials I found on YouTube. I had difficulty with getting the virtualbox to run. Later, I found out that in order to get it to open, I needed to disable “Boot in safe mode” from my system BIOS on my computer. This took a long time to figure out but finally, it was done. Then, I got it up and running and was able to install all of the necessary components in order to run this code. For this problem, I downloaded a folder that had the dataset in it, this was necessary for this assignment.

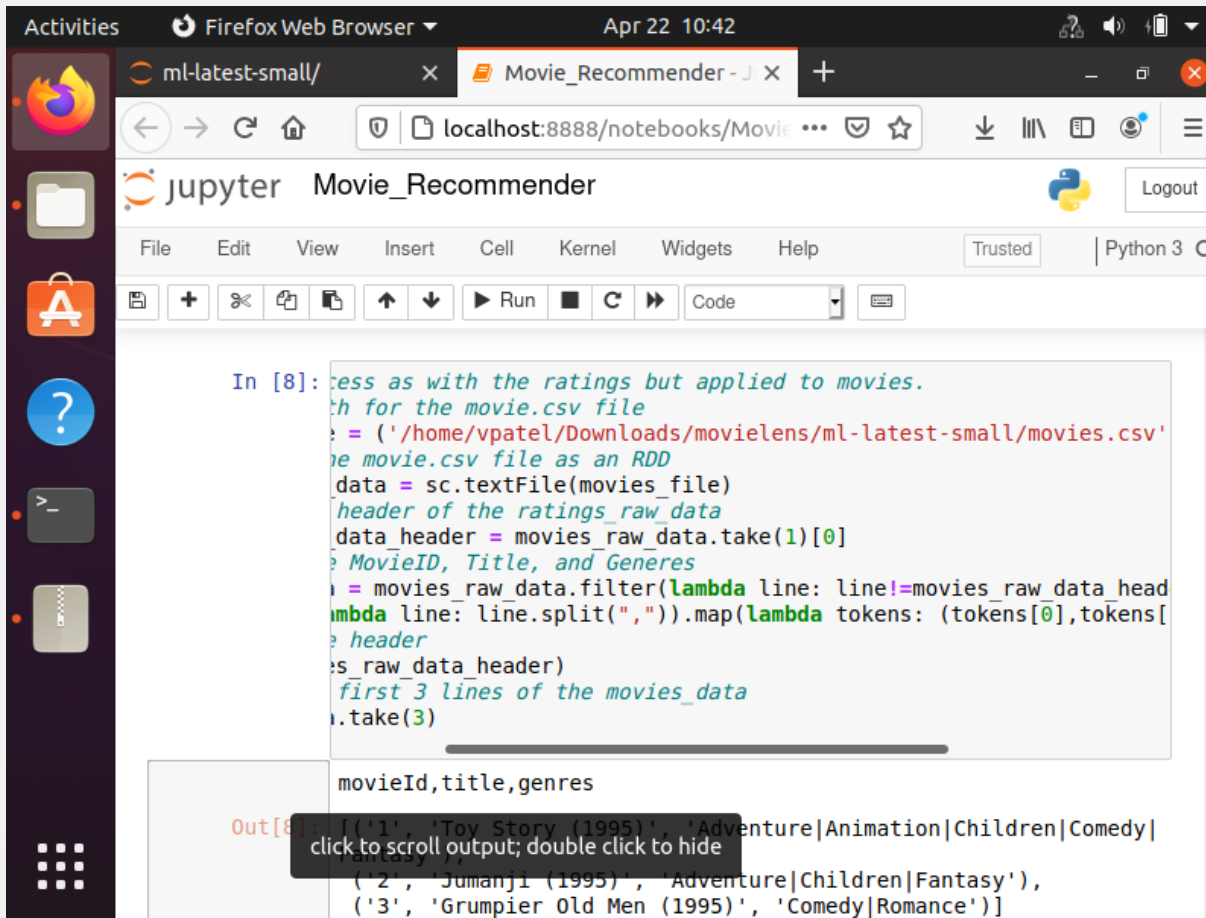
Example of VM running python notebook:



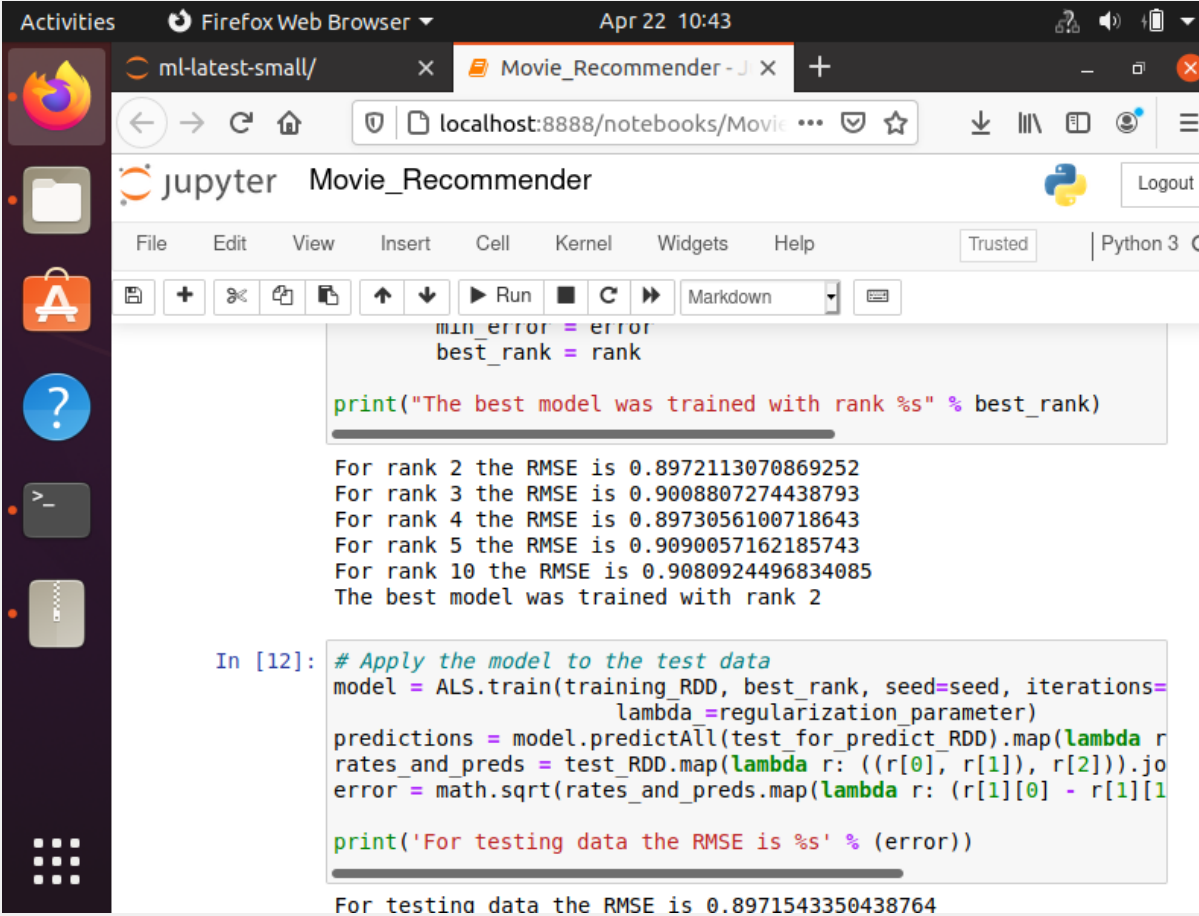
Loading dataset for ratings.csv :



Loading movies.csv dataset:

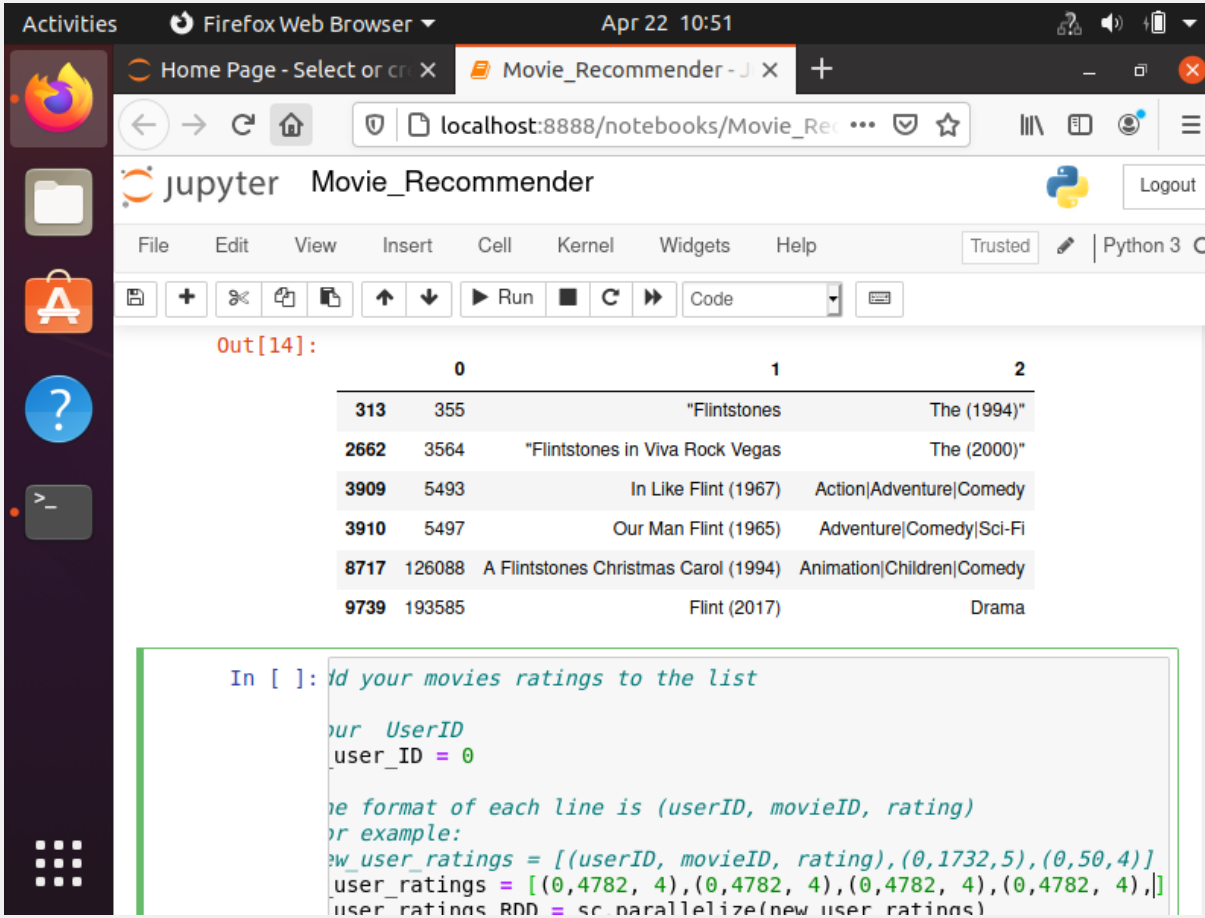


Applying ML Models:

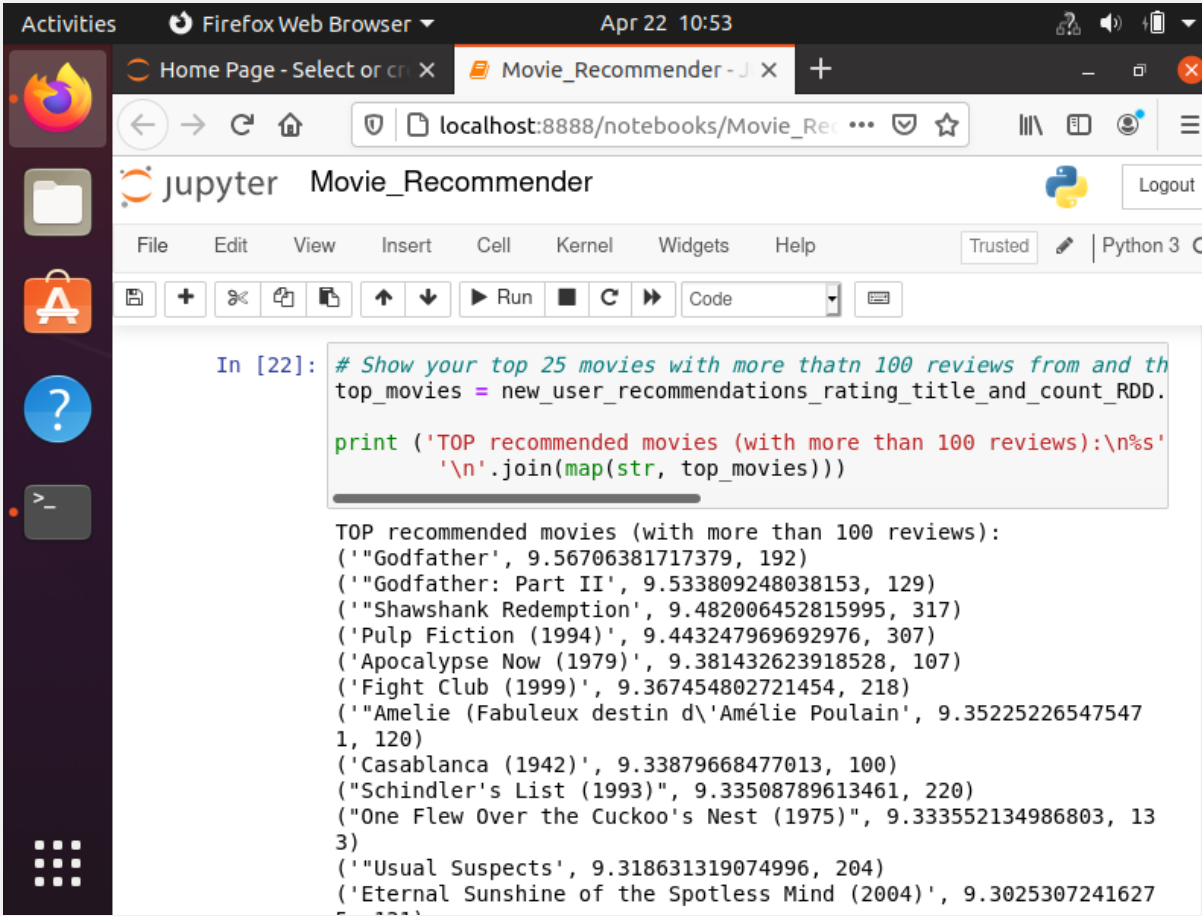


1. **Results**

Finding Movies, adding personal ratings:



Getting Recommendations from Model:



1. **Analysis of Results**

In this lab, I thought that we learned a lot about how to use a model in the real world. Here, we used ALS, MatrixFactorizationModel, and the Rating library. Those libraries were all a part of pyspark. I felt that in this model, there was a very small size of data for the model to really get good results. This was because of user entry, I only entered 5 different reviews of moves that I ranked. I’m sure that if I had more reviews that I had entered, there would be more data for the model to suggest movies off of. I would like to see what number of entries are necessary for good results. This lab had no additional lab questions at the end.

1. **Conclusion**

I thought that this was a really good introduction to something that is very applicable in the real world. When doing this lab, my mind immediately went to apple music, Netflix, Amazon, Spotify, Hulu, YouTube, etc. I felt that all these platforms, and so much more, use something similar to this to suggest videos/content for users to view next. Often, I spend time on YouTube just watching the recommend videos, and before I know it, I have spent hours of my time on there because the recommendations are so good. Obviously, their algorithms are much more complex than this. The positives to this is that it is a real world scenario and I found it really cool how we are doing something that is so similar to what is being done on many platforms already. I didn’t feel like there was any real negative to this lab, other than I did not really understand what role the MatrixFactorizationModel had to play in this. Maybe a more in depth review of what the MatrixFactorizationModel does would have been a little bit helpful. I think that this would be useful in pretty much any field that has content and customers that are viewing and consuming content. I mentioned a few things already that use something like this, I think that this is perfect for more companies such as that. This was my first experience in this type of topic, I thought that it was really interesting how it worked. I would certainly be interested in learning a little bit more about this topic because it is used so widely.

1. **Reference(s):**

Gower, J. C. (1985). Properties of Euclidean and non-Euclidean distance matrices. *Linear Algebra and its Applications*, *67*, 81-97.

Verma, N., Mahajan, D., Sellamanickam, S., & Nair, V. (2012, June). Learning hierarchical similarity metrics. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2280-2287). IEEE.