Comparative Study of Different Collaborative Filter based Recommendation Engines

Submitted by:
Riddhiman Kumar (1729052)
Debasrito Lahiri (1729122)
Hritam Jaiswal (1729130)
Satvik Tiwari (1729155)

Under Guidance from:
Prof. Rajdeep Chatterjee

Contents:

Introduction

- Recommendation Engines have now been in use since quite some time for predicting content for users.
- Common uses include YouTube, Amazon, Netflix etc.
- Based on past user browsing history and reviews of users, new content is recommended to the user based on their personal taste and overall review of the contents.
- It is of various types, the most common being:
  1. *Content based filtering*
  2. *Collaborative filter based*
- Here, we will compare different techniques of collaborative filtering based recommendation engines.
- We will construct the different models and train them on the dataset. Then we will try to predict the user ratings for the remaining data in the dataset.

What is a recommendation engine?

- A recommendation engine is a system that suggests products, services, information to users based on analysis of data.
- The recommendation can derive from a variety of factors such as the history of the user and the behavior of similar users.
- A recommendation engine can significantly boost revenues, Click-Through Rates (CTRs), conversions, and other essential metrics. It can have positive effects on the user experience, thus translating to higher customer satisfaction and retention.
- Let's take Netflix as an example. Instead of having to browse through thousands of titles, Netflix presents us with a much narrower selection of items that we are likely to enjoy. This capability saves us time and delivers a better user experience. With this function, Netflix achieved lower cancellation rates, saving the company around a billion dollars a year.
- Although recommender systems have been used for almost 20 years by companies like Amazon, it has been proliferated to other industries such as finance and travel during the last few years.

Scope and objectives

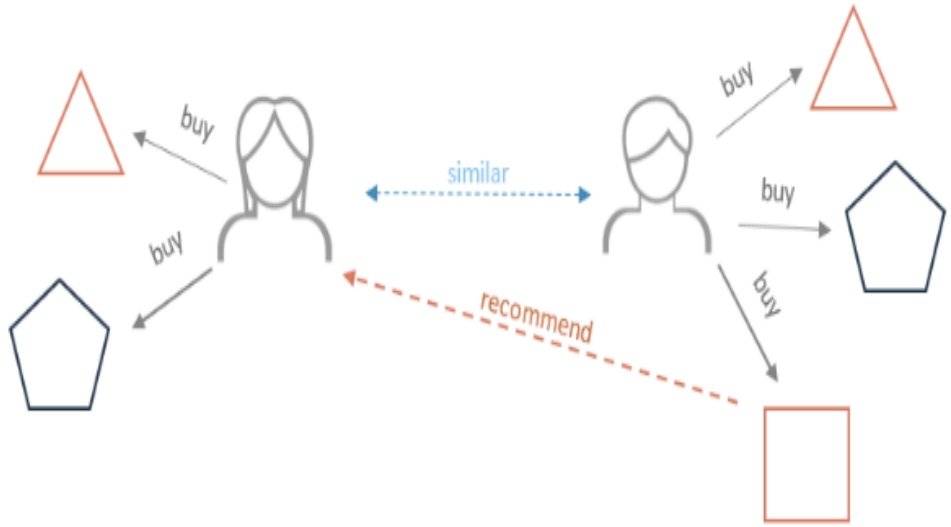
- Providing better and more personalized content to the users.
- More positive feedback from users of these platforms.
- More customer retention for the companies.
- Discovery of new content that the user might not have heard of before but like.
- Increases user engagement in the platform.
- Due to better suggestions spending on the side of the user increases. So revenue of companies increase.
- Acts like a guide to the user through a large collection of content.

Types of Recommendation Engines

- Collaborative Filtering: Collaborative Filtering is a recommendation system technique that collaborates past history of many users to give new recommendations to a specific user.
- Content Based Filtering: Content-based filtering methods are based on a description of the item and a profile of the user's preferences. These algorithms try to recommend items that are similar to those that a user liked in the past, or is examining in the present.
- Multi-Criteria Recommendation Systems: In this system the recommendation is done based on multiple criteria of an item instead of just a single overall preference of an user.
- Mobile Recommender Systems: These are recommendation systems used in mobile devices. The main challenges here are the lack of adequate processing power, noisy data and provision of contextual recommendations. To deal with these problems, most of the processing is done in the cloud and location data is used for contextual awareness.
- Hybrid Recommender Systems: Most recommender systems now combine two or more of the above types to do the recommendations. Ex: Netflix combines both collaborative and content based filtering.
- Session Based Recommendation Systems: These are recommendation engines that do the recommendation based on user interactions in the current session and do not use past data of previous sessions. Ex: YouTube uses such recommendation engines.
- Reinforcement Learning Based Recommendation Systems: In such recommendation systems, the users are the environments while the engines are the agents which act upon these environments in order to receive a reward.

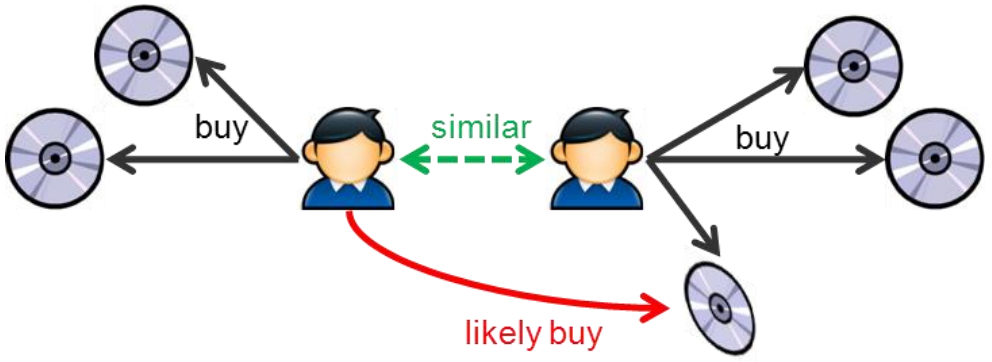
How Collaborative Filters Work

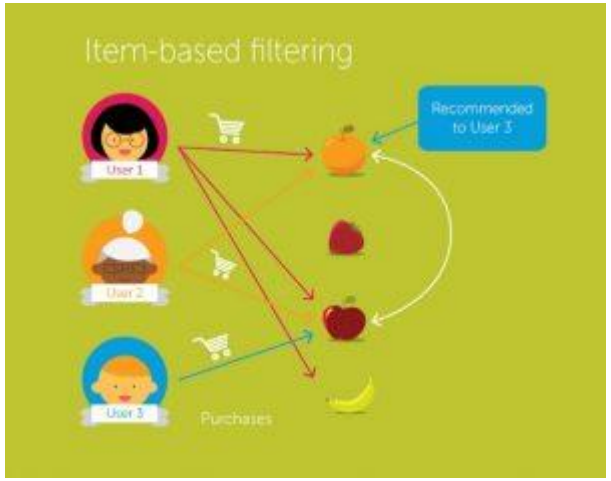
- Collaborative Filtering is a recommendation system technique that collaborates past history of many users to give new recommendations to a specific user.
- The underlying assumption here is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.
- The recommender engine has access to a list of past ratings of different users.
- The current user's past ratings are input to the engine.
- The engine then compares these and gives a recommendation to the user.
- Such comparisons can happen in 2 ways:
  1. Memory Based
  2. Model Based



Memory Based

- This method uses a rating system to predict the preferences of one user by taking into consideration the preferences of a similar user, or the ‘neighbour’.
- It is of 2 types, user based and item based.
- User based: Here, the existing ratings of an user A is matched with other users with similar ratings for the same items. Then, for the items which have not been rated yet by A, the ratings of these other set of users for these items are considered to predict the likely rating to be given by A for these.
- Item based: In Item-Based Collaborative Filtering, we compare two items and assume them to be similar when one user gives the two items similar ratings. We then predict that user’s rating for an item by calculating the weighted average of ratings on most X similar items from this user.





Memory Based

- Here we will use cosine similarity matrix to create the recommender engine.
- Cosine similarity is a mathematical method to find the similarity between 2 items.
- It is the angle in between the spatial distribution vectors of the items and is given by:

$$\cos\theta = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|\|\vec{B}\|}$$

Where,

$\vec{A}\ and\ \vec{B}$ =Vectors representing the 2 items and are a collection of their numerical features

θ = The angle between the vectors.

- The less the value of the angle the more similar the items are.
- A matrix representing all such cosine similarities between all the items with respect to each other is called a cosine similarity matrix.
- The engines creates such a matrix for the user/item showing the similarities in ratings by different combinations of users for the same movie/different combinations of movies for the same user.
- Then the engine uses KNN algorithm to predict the closer rating(in item based)/closer user(in user based) based on the cosine similarity matrix data.
- In case of user based, the ratings of this closer user is then considered for predicting the rating and for item based the closer item is considered for predicting the rating.

Model Based

- In this method of collaborative filtering recommender systems, different data mining and machine learning algorithms are used to develop a model to predict a user's rating of an unrated item.
- Some examples of these models are Bayesian networks, clustering models, singular value decomposition, probabilistic latent semantic analysis, multiple multiplicative factor, latent Dirichlet allocation and Markov decision process-based models.
- Such models find trends in the ratings being given by users to predict the rating an user will give to an unrated item.
- Here we will use singular value decomposition of matrices.
- It is a decomposition of a rectangular matrix A of dimensions mxn where,

$$A = USV^T$$

Here,

A= The original mxn matrix

U= an mxm orthogonal matrix with eigenvectors of AA^T

S= an mxn diagonal matrix with eigenvalues of both $A^T A \; and \; AA^T$

V=an nxn orthogonal matrix with eigenvectors of $A^T A$

- Using SVD we will factorize the matrix of users and ratings into 2 separate matrices of ratings and users.
- These matrices contain vectors products of which model an user's interaction with an item.
- This model will then be used for predicting new ratings.

Project breakdown: Part 1

- Importing of libraries
- Importing of dataset and extracting the ratings data from it
- Dividing the dataset into 2 parts

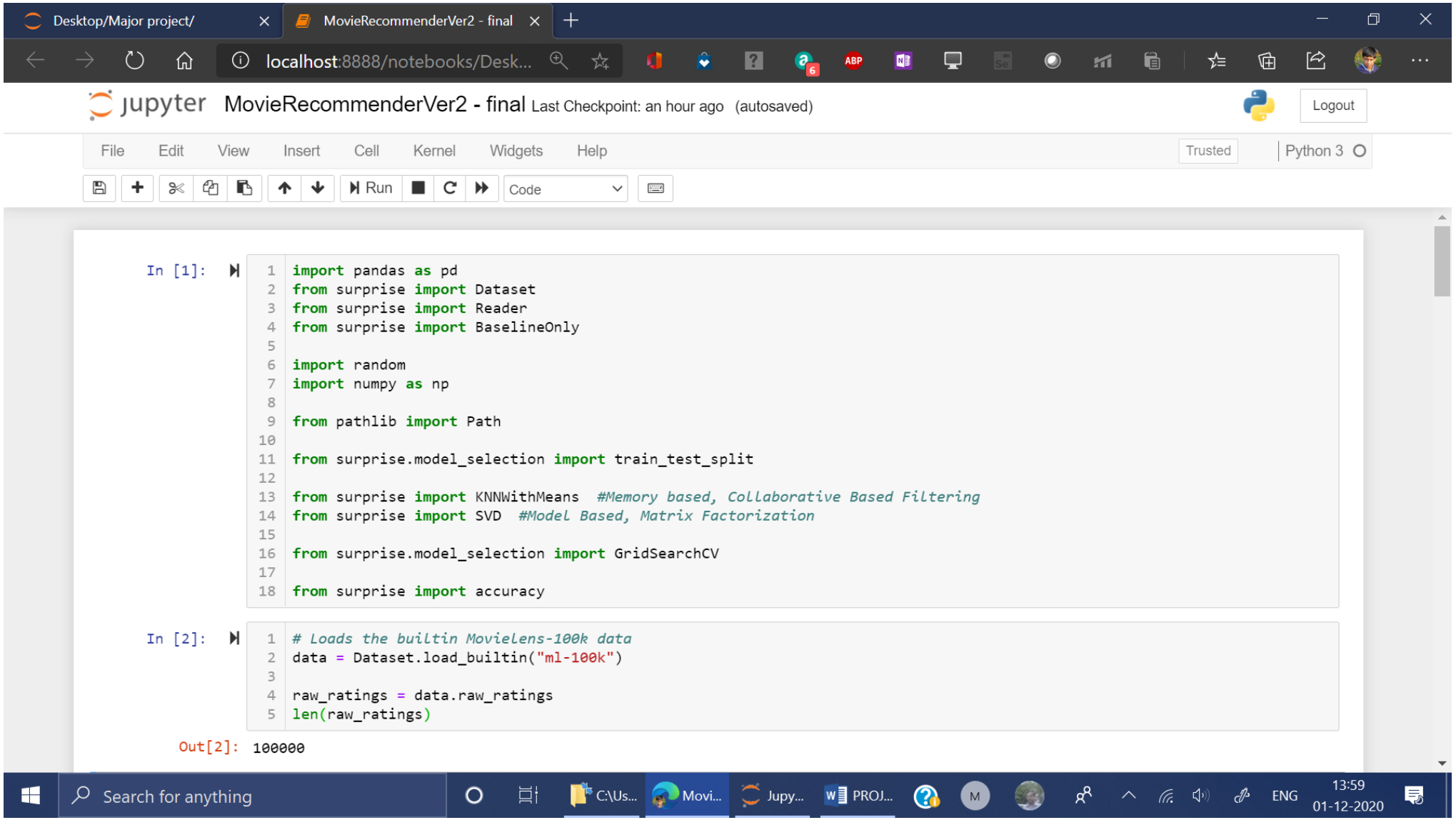
Project breakdown: Part 2

- Setting the hyperparameters for GridSearchCV using KNN algorithm
- Fitting the smaller dataset part in the model
- Finding the best score and set of parameters for the model
- Building the training dataset using the larger part of the dataset and the test dataset using the smaller part.
- Training the model using the training dataset
- Do predictions using the trained model and the test dataset
- Find RMSE, MSE and MAE for Memory based

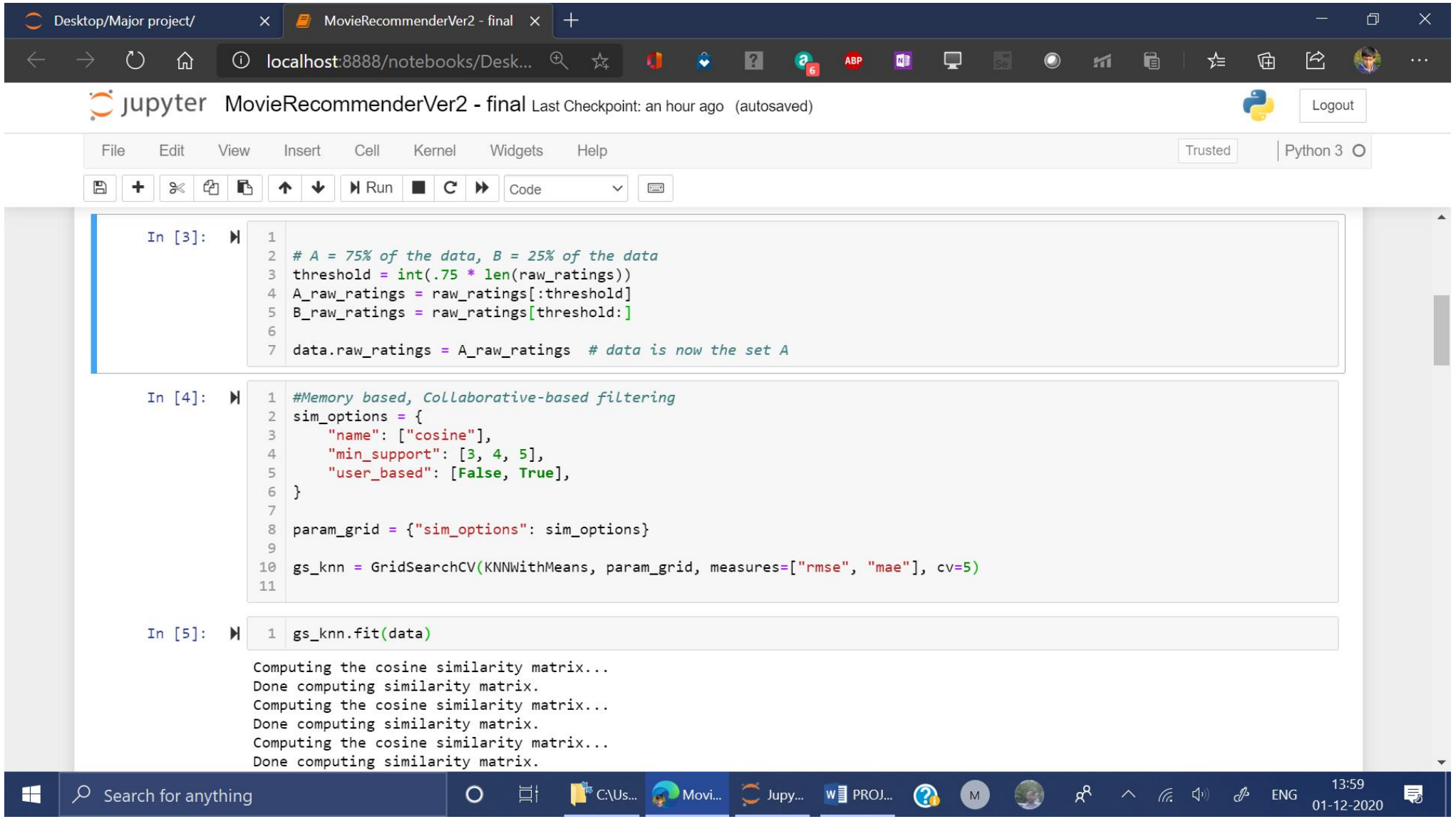
Project breakdown: Part 3

- Setting the hyperparameters for GridSearchCV using SVD
- Fitting the larger dataset part in the model
- Finding the best score and set of parameters for the model
- Building the full training dataset and the test dataset using the smaller part.
- Training the model using the training dataset
- Do predictions using the trained model and the test dataset
- Find RMSE, MSE and MAE for Matrix factorization based

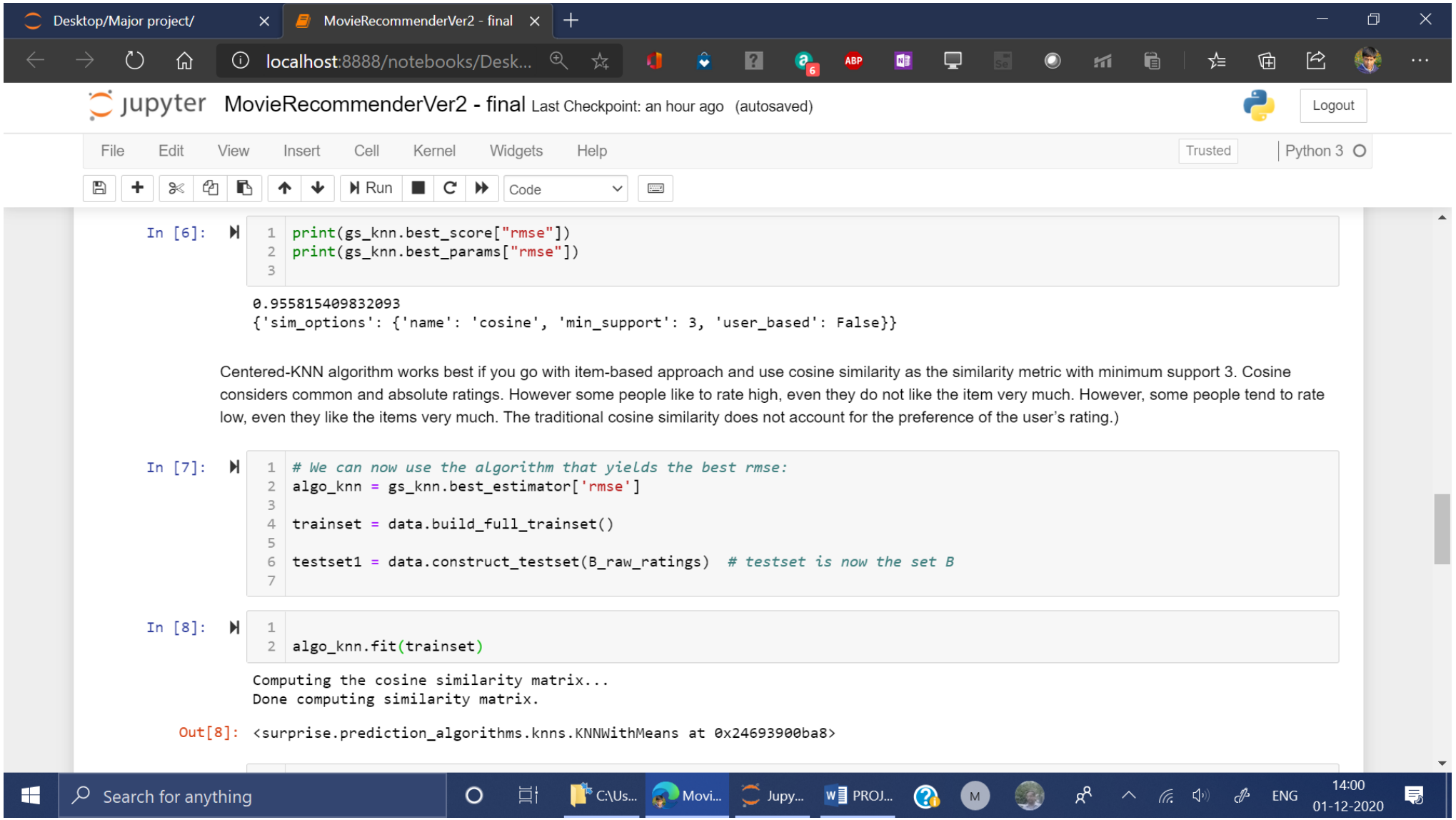
Implementation



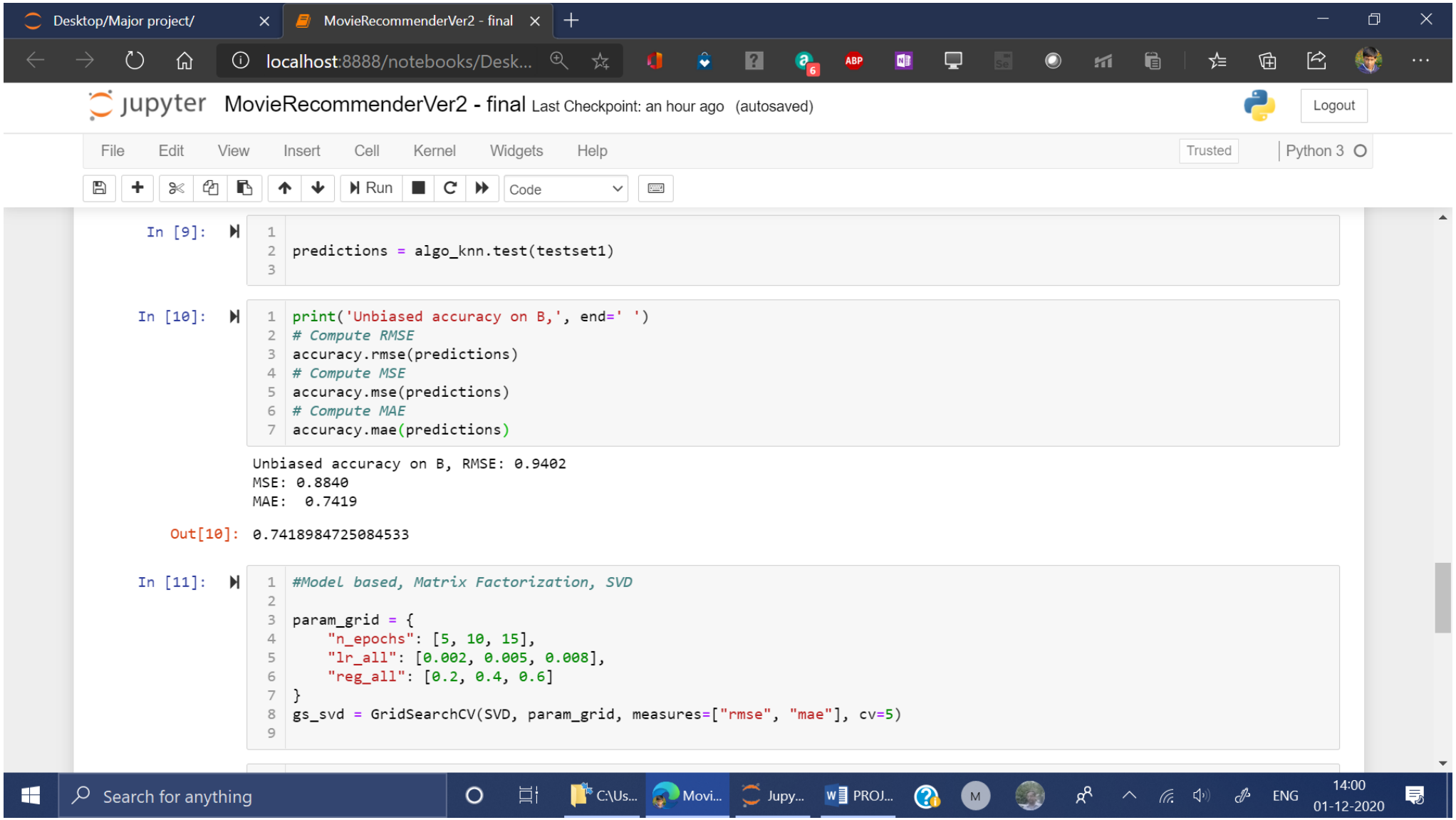
Implementation



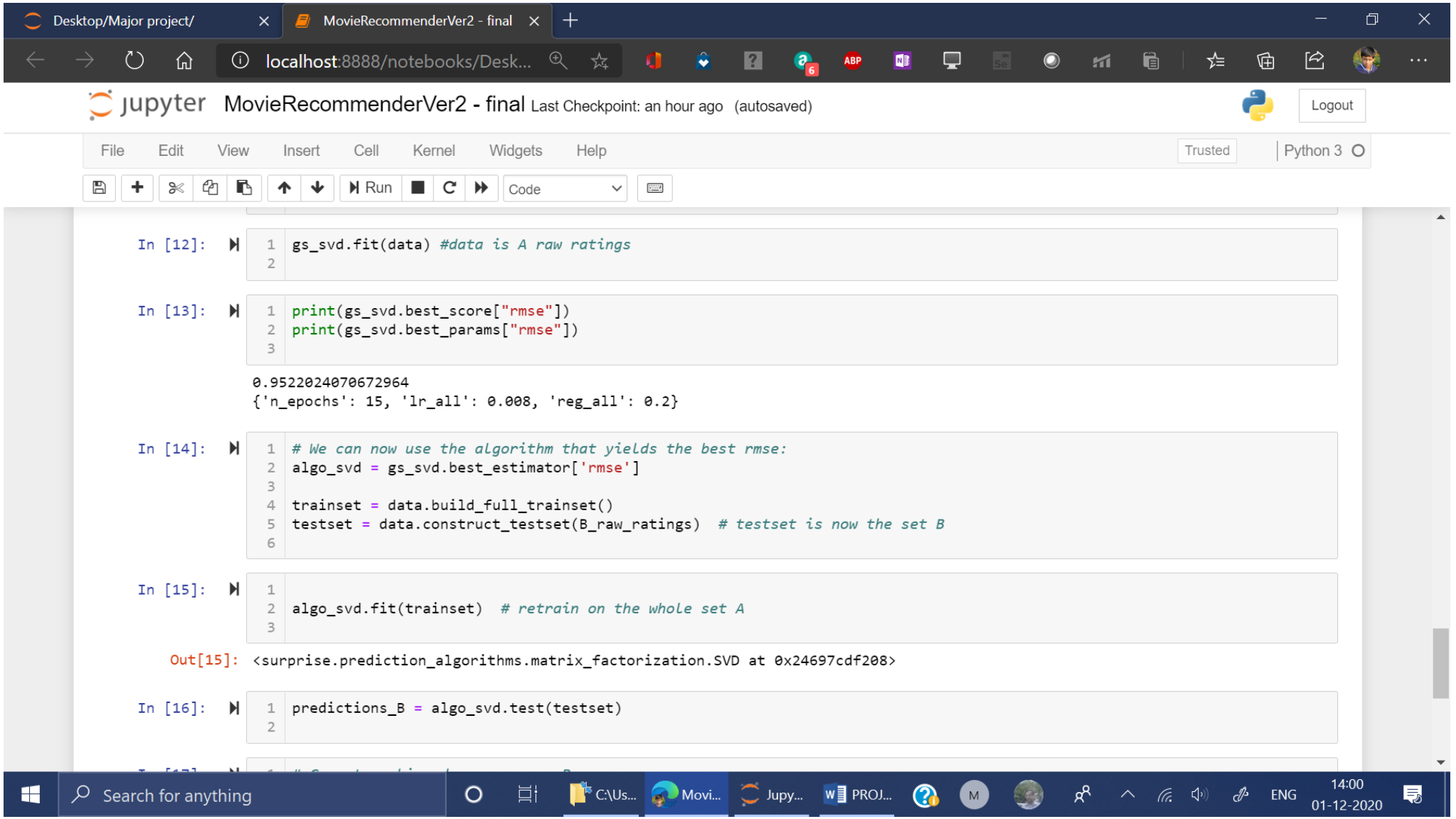
Implementation



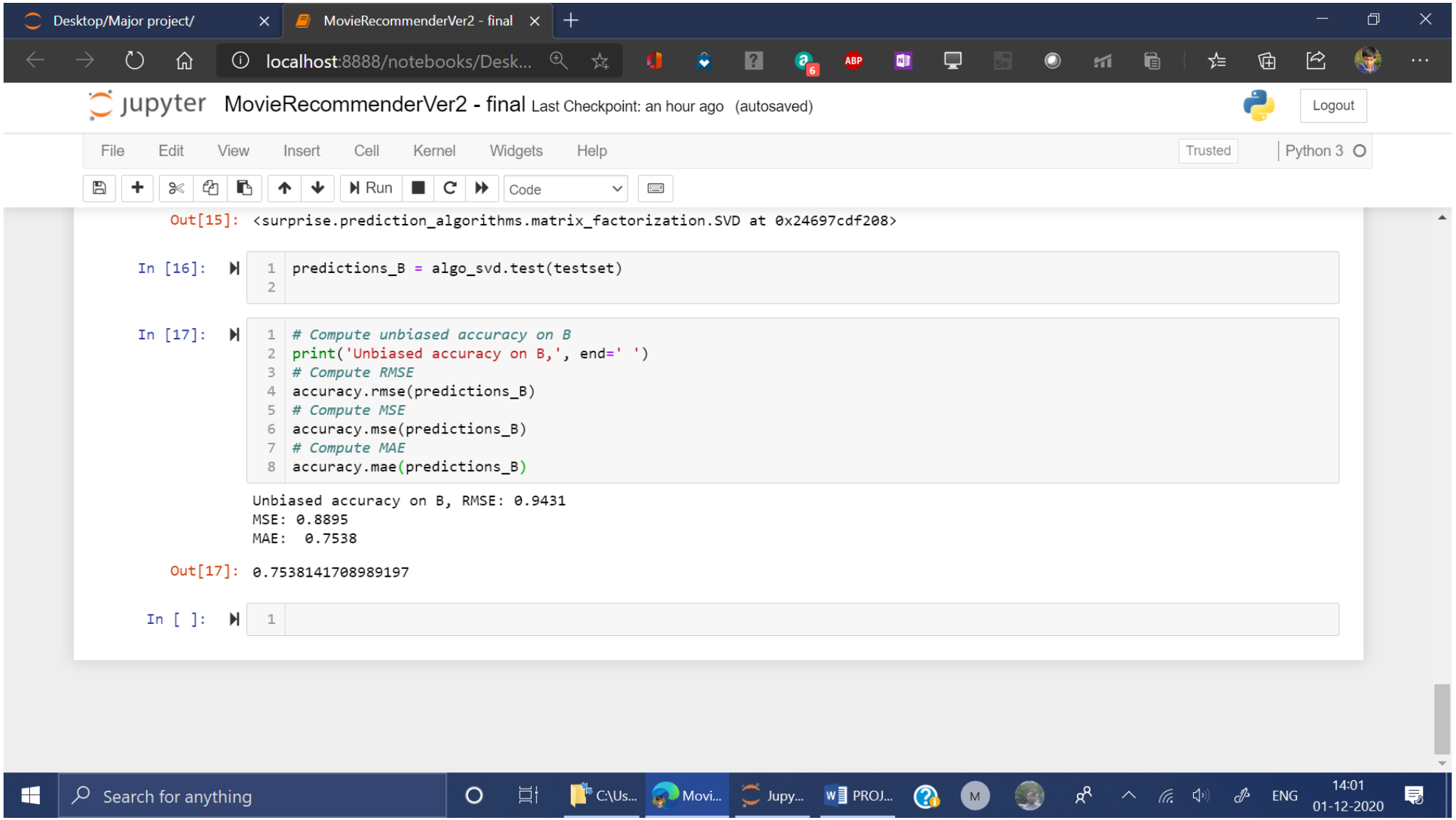
Implementation



Implementation



Implementation



Observations

- The following metrics were obtained in the different filters:

Metric	Memory Based	Model Based
MAE	0.7419	0.7538
MSE	0.8840	0.8895
RMSE	0.9402	0.9431

- As we can see, for every metric both the models give almost similar amounts of errors. The memory based filter only performs slightly better which is almost insignificant.
- This shows that both the filters are equally efficient in predicting the user ratings.
- Also, it should be noted that the error is always less than 1.
- This means that the models can predict the user ratings with a maximum error which is less than 1.
- But considering that the ratings are between 1 and 5 and are only integer values, the model should effectively predict the correct rating without any error in most cases.
- The only time there will be an error is when the number of users to predict for is very high and the average rating of all those users are considered for evaluation.

Conclusion

- We thus constructed 2 recommendation engines of collaborative filter type and did predictions using it while checking the performances of these engines
- We found out that the recommendation engines give a maximum error of ≈0.75 for MAE and ≈0.94 for RMSE i.e. both the errors are <1.
- This shows that such recommendation engines are fairly accurate and effective in performing recommendation for users provided the input data given to the engines are cleaned and processed properly.
- Also, the value of these errors can be reduced significantly further by using neural networks for our recommendation engines.
- Hence, such recommendation engines can be used by companies for better predictions and to provide more personalized content for users which will in turn increase customer satisfaction and sales of the company.

References

1. https://en.wikipedia.org/wiki/Recommender_system
2. https://en.wikipedia.org/wiki/Collaborative_filtering
3. http://mlwiki.org/index.php/Singular_Value_Decomposition
4. https://medium.com/acing-ai/what-is-cosine-similarity-matrix- f0819e674ad1
5. https://en.wikipedia.org/wiki/Cosine_similarity
6. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm
7. https://www.mygreatlearning.com/blog/gridsearchcv/
8. http://surpriselib.com/

Thank You!