2020

Alfred D. Hull

Springboard

3/11/2020

Recommender Systems Final Report

[](https://github.com/ahull002/P001.recommender.systems) [](https://www.linkedin.com/in/alfredhull/) [](https://www.springboard.com/)

# **Table of Contents**

[**Table of Contents** 1](#_Toc35178624)

[Phase I. Business Understanding 2](#_Toc35178625)

[Background 2](#_Toc35178626)

[Set Objective 0](#_Toc35178627)

[Cost and Benefits 0](#_Toc35178628)

[Constraints, Limitations, and Assumptions (CLAs) 2](#_Toc35178629)

[Phase II. Data Understanding 3](#_Toc35178630)

[Initial Data Collection Report 3](#_Toc35178631)

[Data Description Report 3](#_Toc35178632)

[Exploratory Data Analysis (EDA) 4](#_Toc35178633)

[Data Quality Report 6](#_Toc35178634)

[Phase III. Data Preparation 7](#_Toc35178635)

[Rationale for inclusion/exclusion 7](#_Toc35178636)

[Data Cleaning 7](#_Toc35178637)

[Phase IV. Data Modeling 9](#_Toc35178638)

[Select Modeling Technique 9](#_Toc35178639)

[Generate Test Design 9](#_Toc35178640)

[Data Sets to be Modeled 9](#_Toc35178641)

[Variance Inflation Factor (VIF) 9](#_Toc35178642)

[Build Model 10](#_Toc35178643)

[Logistic Regression 10](#_Toc35178644)

[Phase V. Model Evaluation 11](#_Toc35178645)

[Interpreting The Results 11](#_Toc35178646)

[Odds Ratio 12](#_Toc35178647)

[Phase VI. Deployment and Conclusion 13](#_Toc35178648)

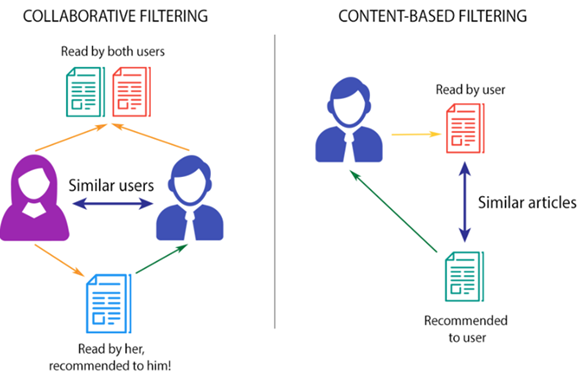
[Looking Forward 13](#_Toc35178649)

[Increase the Number of Observations 13](#_Toc35178650)

# Phase I. Business Understanding

## Background

As of 2019, we have found ourselves at a new precipice in entering the Fourth Industrial Revolution: The Real-Time Enterprise! Just like the intersections of the past: First in 1784, Water and steam; Second in 1870, First conveyor belt; and the Third in 1969, Electronics and information technology, this pause will bring with it both challenges and new opportunities. Data stewards in the discipline have realized associative costs for data storage has expounded this issue while driving talent gaps. As organizations continue to ingest voluminous amounts of data, they must become more tactical with the data they are creating and using to sustain or improve Return on Investment (ROI). More so, the approach of advancing insight through analyses of mountains of data must be clear and consistent so that results are impactful, insightful, but more so reproducible.

Given the current shortfall of talent proportionate to the opportunities seen in mining volumes of data and the time-consuming aspect in discovering the best possible options, it is ever more critical that machine intelligence and automation play a progressive role within organizations. An extremely effective way of optimizing selection in a sea of choice is by deploying recommender systems. Recommender Systems are capable of predicting the future inclination of a set of items for an individual, team, or even entire organization, then optimizing selection based on commonalities of historical transactions. One fundamental value propositions of a recommender system in modern society is that it enables people to maximize choice/selection due to an onslaught of varieties with the prevalence of the Internet and a decreasing cost of data storage. Imagine going into your local library and searching for a cookbook. The librarian tells you that there is an entire wing of the library dedicated to cookbooks: have fun. While the information collected in the library may sound impressive, the hunt for what you may be looking for will undoubtedly be exhausting. If there was a way to provide insight about your needs given your comfort level of sharing your data, learning algorithms could help match your ideas with others who are similar then prescribe analogs that may better suit your needs.

This project progresses logically through a recommender analysis utilizing the Cross-Industry Standard Process for Data Mining (CRISP-DM). The aim is enabling data stewards and stakeholders to clearly understand what, when, where, why, and how the Data Science work proceeds. The phases include Business understanding, Data Understanding, Data Preparation, Modeling & Application Development, Evaluation, Model Deployment.

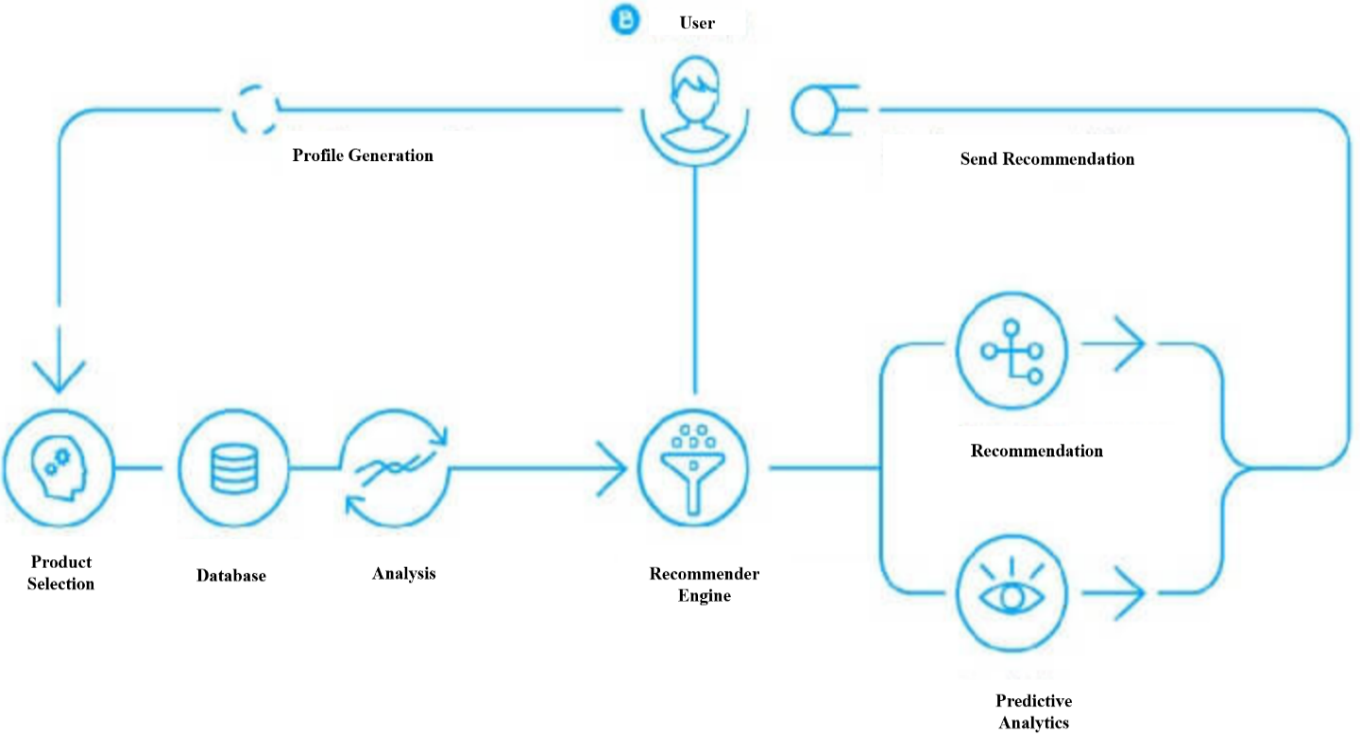
## Who might care?

Maintenance Managers, Operations Managers, Knowledge Management Teams, Human Resource Departments in organizations, and especially tech organizations such as Microsoft, Amazon, Google, Apple, and individual agencies within the Department of Defense, etc. can use such a model to better market products and or materials to customers.

## Set Objective

The objective here is to build a ***Model-Based: User-Item collaborative filtering*** that will recommend the probability from 0 to 1 of users liking a particular movie based on the history of other users who have liked the same film wants on other videos. The Singular Value Decomposition (SVD) algorithm will be used to predict users' ratings on unrated items.

As users log into an entertainment platform and want to watch a movie: the recommendation system will need to suggest the movie rating of the user based on other similar users who have watched this movie also. The system will also compare the ranks of other users who have viewed similar videos to enhance the pushed selections. The business reasons to do this are many, and the important ones are:

1. ***Relevancy***. Movie ratings must be relevant for the user, taking into consideration the user's and other users' past ratings and interests. Additionally, due to the vast amount of choices available, this recommender will aide users in siphoning through massive amounts of options in a blink of an eye.
2. ***Customer Retention.*** The system must put in its best efforts to keep its customers. For example, if the system recommends a movie to a user, and the customer liked it, they may continue coming back for more content.

## Cost and Benefits

While the benefits of recommendation systems are exponentially impactful to an organization, fundamentally shifting the labor costs associated with marketing re-targeting efforts by automation. The system development is not natural to guess from a top-level view. It requires a personalized approach and thorough understanding and analysis of a client's business processes, set-objective, and data. Roughly the cost for this would range from ***$25,000.00 to $45,000.00***. One way to build a rough order magnitude to factor costs is by looking at such a project through four key areas:

1. **Analysis and rough order magnitude (ROM) estimation – $5,000.00 (40 hours burdened)**

* ***Business Understanding***. Preliminary to the project start date, it is critical to set up set-objectives (business-understanding). The primary goal of this phase is to work with business analysts to conduct a feasibility study. The work on every project starts with this analysis.
* ***Data Understanding***. Here we analyze currently used metrics, data assets, and customer set-objectives and business processes.
* ***Data Preparation***. In this phase, the team defines growth points, determines the appropriate technological stack, timelines, and budgets, while developing all corresponding documentation with wrangled data.

1. **Prototype development – $5,000.00 - $10,000.00 (40 to 90 hours burdened)**

* ***Data Modeling***. During this phase, we develop a first-go prototype of the recommendation system based on the data gathered during the previous step (Data Preparation) to test the hypothesis and show its efficiency. The Business and Data Understanding Phases are critical to calculating any final costs of AI-powered recommender development.

1. **Minimal Viable Product (MVP) development (with prototype included) – $10,000.00 - $15,000.00 (90 to 140 hours burdened)**

- ***Model Evaluation***. The previous step was just a simple prototype only and not the alpha version of the recommendation system. Here we will evaluate the alpha version for deployment.

The main difference between Prototype Development and Minimum Viable Product (MVP) is as follows. The first proves the concept or idea –to extract insights from the data, and according to those insights, generate suggestions and provide recommendations; MVP, on the other hand, is the original alpha version (feature testing) of a product that solves the set-objective.

1. **Deployment and Release – $5,000.00 – 15,000.00 (40 to 140 hours fully burdened)**

* ***Deployment & Conclusion***. During the last phase, improve the MVP of the recommendation engine to match the customers' needs and integrate it within any existing infrastructures.

The final Release can, in some cases, be delayed or even postponed according to various factors. Often we choose agile software development and continuous integration/continuous development (CI/CD) and prefer to launch the recommender prototype, improving it over time while in production.

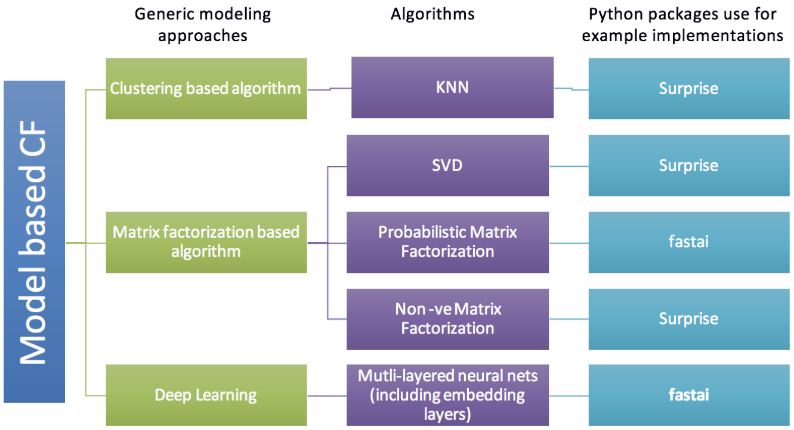
## Constraints, Limitations, and Assumptions (CLAs)

There are two sorts of recommender engines: ***Memory-based*** and ***Model-based*** methods. The most common sorts of recommendation systems are ***content-based*** and ***collaborative filtering***. In collaborative filtering, the behavior of a cluster of users form recommendations to other users. The actual recommendations predicate the preference of other users. For example, recommending a movie to a consumer because their friend also liked the movie. The simplest to build and deploy are memory-based techniques. The advantage of *memory-based* technologies is that they're simple to implement, and therefore the resulting recommendations are often both easy to communicate and evaluate.

**Memory-Based methods**

1. **User-based collaborative filtering** suggests products to a consumer based on the very fact that other users like the products with a high degree of commonalities.
2. **Item-based collaborative filtering** identifies similar items that supported users' previous ratings. For example, Consumer A, B, and C gave a thumbs-up rating to books X and Y. The next user, D buys book Y; they get a recommendation to purchase book X. This is because the model identifies book X and Y as being similarly supported by the ratings of users A, B, and C.
3. **Content-based methods.** Use metadata like genre, producer, actor, musician to recommend items say movies or music. For example, you were supporting Infinity War that featured Robert Downey Junior because someone watched and liked Iron Man. Similarly, you'll get music recommendations from individual artists because you liked their music. Content-based systems support the thought that if you wanted a particular item, you're presumably to love something that's almost like it.

**Model-based methods**

Are established using data mining, machine learning algorithms to predict users' ratings on unrated items. During this approach, techniques such as dimensionality reduction improve model accuracy. Examples of such model-based methods include decision trees, rule-based models, Bayesian methods, and latent factor models (Mwiti, D. (2019, December 3)).

# Phase II. Data Understanding

## Initial Data Collection Report

The dataset was put together by the Grouplens research group at the University of Minnesota. It comprises 1, 10, and 20 million ratings. And can be found at <https://grouplens.org/datasets/movielens/>. The following Data Collection Report is a simple listing of the data sources acquired along with their locations, the procedures used to procure them, and any difficulties encountered. This section will aid both with future replication of this project and with the execution of similar future projects.

**Data Collection:**

1. **Data Source:** MovieLens (
2. **Location**: <https://grouplens.org/datasets/movielens/>
3. **Movie Ratings**: Small: 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users. Last updated 9/2018
4. **Method**: This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 classifications and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996, and September 24, 2018. This dataset was generated on September 26, 2018.
   1. *Users were selected at random for inclusion.*
   2. *All selected users had rated at least 20 movies. No demographic information is included. An id represents each user, and no other information is provided.*
   3. *The data are contained in the files links.csv, movies.csv, ratings.csv, and tags.csv. More details about the contents and use of all these files follow.*
   4. *This data set and other GroupLens data sets are publicly available for download at http://grouplens.org/datasets/.*
5. **Obstacles**: This is a recurring developmental dataset. As such, it may change over time and is not an appropriate dataset for shared research results. See available benchmark datasets if that is your intent.

## Data Description Report

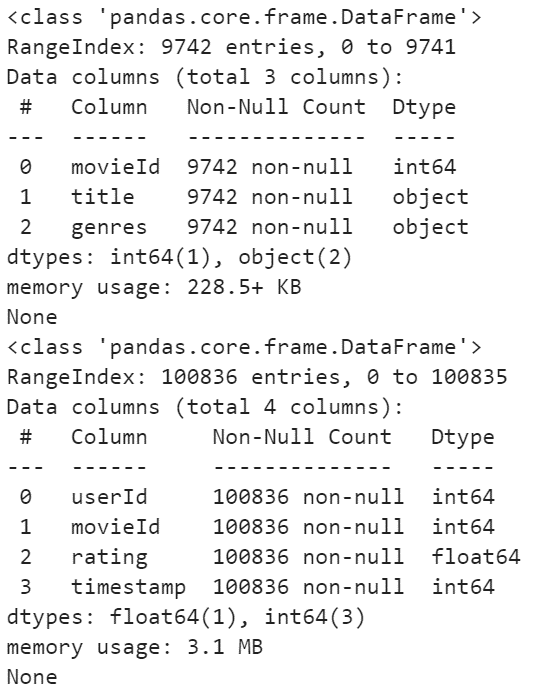
*This report describes the data's type, quantity, characteristics, and any other surface physiognomies discovered.*

1. **README**: <http://files.grouplens.org/datasets/movielens/ml-100k-README.txt>
2. user\_id - the ID of the user who rated the movie.
3. item\_id - the ID of the film.
4. rating - The rating the user gave the movie, between 1 and 5.
5. timestamp - The time the film was rated.
6. title - The title of the movie.

## Data Quality Report

*This report lists the results of the data quality verification along with solutions vetted by subject matter experts to validate any concerns by assessing the following questions: Is the data complete and accurate; does it encompass errors and how popular are they?*

As we can see with the information printout from pandas, the dataset is rather clean (complete & accurate) and has already been prepped for: Exploratory Data Analysis (EDA) and Modeling. For instance:

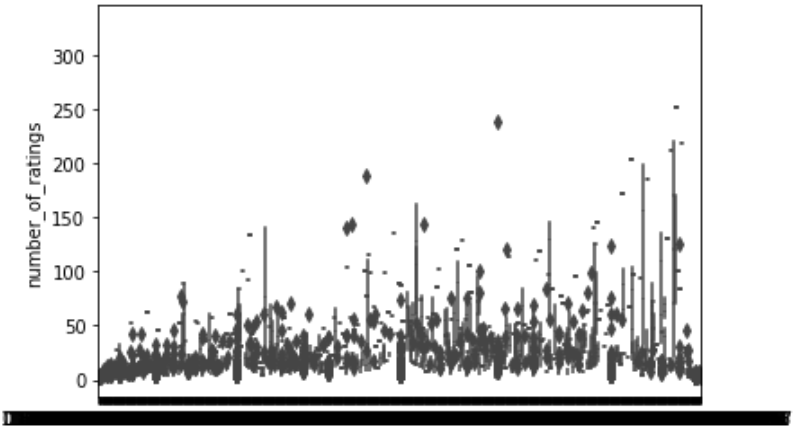


1. There were 9,742 observations recorded that are of a consistent data type for movies.
2. There were 100,836 observations recorded that are of a consistent data type for ratings.

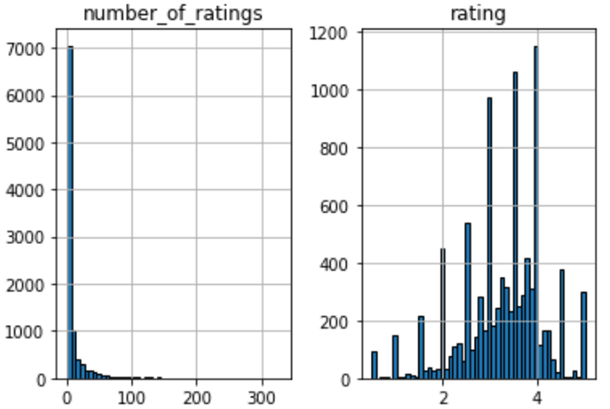
## Exploratory Data Analysis (EDA)

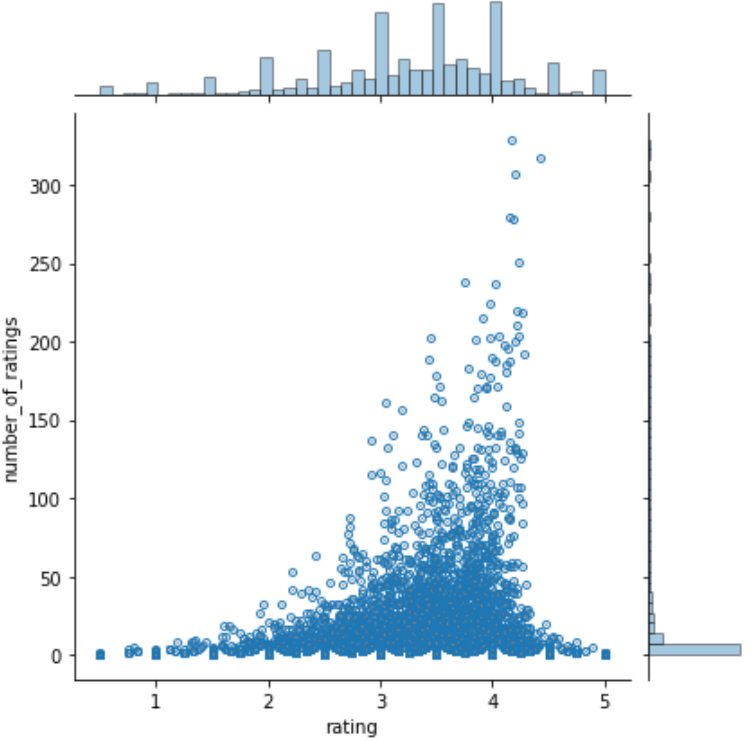
EDA describes the results of exploring the data involved in this project. EDA includes the first findings and or initial hypothesis and their impact on the remainder of this project.

**Data Visualization and Interpretation:**



The graph above shows that an excellent safe point for cutting off outliers is around the number\_of\_ratings mark at about 50. This graphic shows that anything higher than this number may skew the recommender's results due to popularity and an abnormal amount of ratings.

Furthermore, the graph below shows that most people rated only about 0-5 movies while also rating them on average 2 to 4. From the sets of histograms, it is clear that most films were not frequently rated. Movies with most ratings are those that are most popular.

To go a step further, the next graph to the left highlights that high numbers of ratings are highly skewed with a high frequency of scores. From the diagram, we can see that there is a positive relationship between the average rating of a movie and the number of ratings. The graph indicates that the higher number of ratings a film gets, the higher the average rating it also gets. This is important to note, especially when choosing the threshold for the number of ratings per movie.

# Phase III. Data Preparation

## Rationale for inclusion/exclusion

N/A

## Data Cleaning

N/A

# Phase IV. Data Modeling

## Select Modeling Technique

### Data Sets to be Modeled

1. For this project, we will build a simple item-based recommender system. To do this, we will need to convert our dataset into a matrix with the movie titles as the columns, the user\_id as the index, and the respective ratings as the values. By doing this, we shall get a data frame with the columns as the movie titles and the rows as the user ids. Each column represents all the ratings of a movie by all users. The score appears as NAN, where a user didn't rate a particular movie. We shall use this matrix to compute the correlation between the ratings of a single film and the rest of the film in the model. We will utilize the pandas pivot\_table utility to create the movie matrix.
2. All Features

## Build Model

All code is available in the Github repository complementing this report.

Singular value decomposition is a method of decomposing a matrix into three other matrices:



**Where:**

A is an m × n matrix

U is an m × n orthogonal matrix

S is an n × n diagonal matrix

V is an n × n orthogonal matrix

# Phase V. Model Evaluation

Here I will interpret the models according to my domain knowledge, the data mining success criteria, and the anticipated test design. I will also decide on the success of the application of the modeling and discovery methods technically, may contact concerned business analysts and domain experts later to discuss data mining results in the appropriate business perspective. While this task considers models, the evaluation phase (Phase 5) further considers all other effects that occurred in the course of this analysis.

At this stage, it is also important to rank the constructed models while assessing them according to any evaluation criteria. I will take the business objectives' success criteria into account as far as possible here.

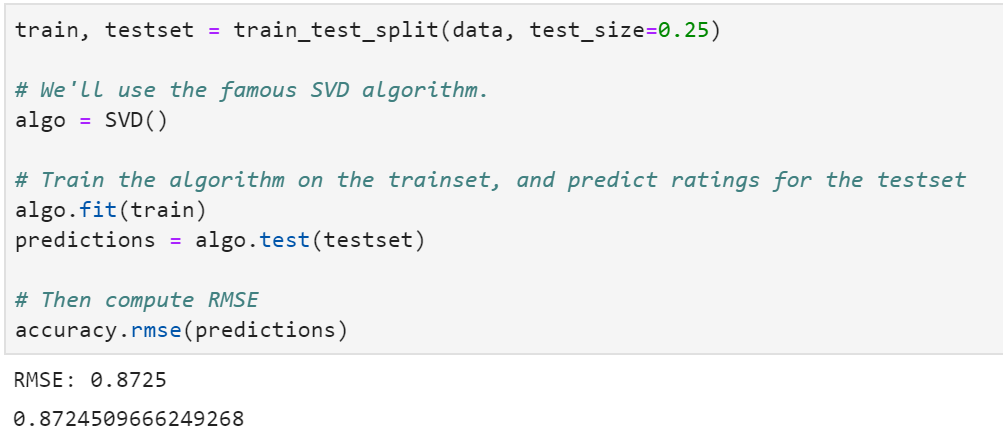
## Interpreting The Results

A matrix was developed: rows as movies and columns as user ids. The essence of SVD algorithm is that it decomposed this matrix into a product of 3 matrices with nice mathematical properties:

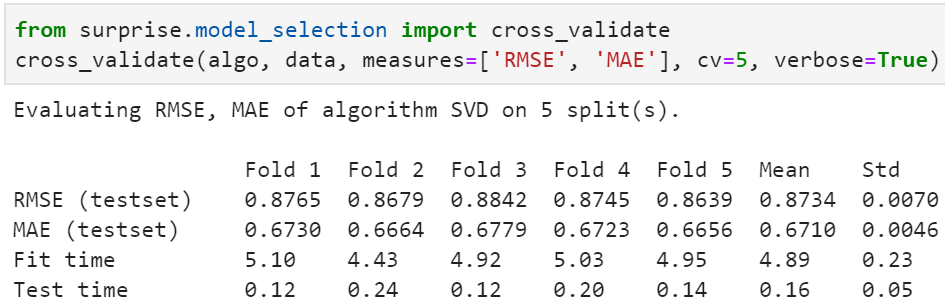
A=USVT.

By analogy, any number can decompose into 3 additional numbers to always have the smallest prime in the middle. E.g 24 = 3×2×4 or 57 = 1×3×19.

The result of the decomposition above left us with an ordered matrix of singular values which include the variances associated with every direction. We assume that larger variances mean less redundancy and less correlation and translate more structure about the data. This permits us to use a representative subset of user rating directions or principal components to recommend movies using this algorithm in the model. When applying the algorithm against the data we have received an 87% accuracy.



When testing these results during a k-Fold Cross-Validation: resampling to evaluate the above machine learning model we also achieved on average the same accuracy response as seen below!



# Phase VI. Deployment and Conclusion

Primarily this model-based recommender from a surface level appears to work very well. I have to blind tested this on ten friends by asking them their favorite movie, re-running the model on it, and recommending other films to them. It appears that the wisdom of the crowd's aspect works well as with the more observations you have in building the correlation matrix, the more outliers get washed out. Furthermore the SVD component proved positive showing that the training data does in fact facilitate enough observations to positively predict future ratings of new users as they appear.

## Looking Forward

### How to improve this recommender system

This system could be enhanced by building a Memory-Based Collaborative Filtering based system. In this case, we'd divide the data into a training set and a test set. We'd then use techniques such as cosine similarity to compute the similarity between the movies.

### Further Research

There are other techniques for building recommender systems. Deep learning is one of the ways of doing so, especially when you have massive datasets. Different algorithms deployed currently to build advanced recommender systems include Autoencoders and Restricted Boltzmann machines.

Reference:

Mwiti, D. (2019, December 3). How to build a Simple Recommender System in Python. Retrieved October 13, 2019, from <https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d>

Deng, H. (2019, December 5). Recommender Systems in Practice. Retrieved October 13, 2019, from https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a