**Interim Project report on**

**Marketing and Retail Analytics**

**“Building Recommendation engine for e-commerce business”**

**Group No. 1**

Batch: PGP-BABI April 2019

Location: Bengaluru

**Group Members**

1. Anupam Verma – PGPBAB19042

2. GVD Shree Harsha – PGPBAB19012

3. Shiv Kumar – PGPBAB19068

4. Shyam Sampath – PGPBAB19055

5. Swarup Ghadei – PGPBAB19019

**Research Supervisor - Vipin Garg**

**Great Lakes Institute of Management**



**Contents**

1. Introduction

2. Scope problem statement and Objective

3. Data Source and Description

4. Analytical Approach

5. Data Pre-processing

6. Exploratory Data Analysis

7. Model approach

8. Insights and Findings from Exploratory Data Analysis

9. Future course of Action

10. References and Bibliography

11. Appendix

1. **Introduction**

The objective of our project is to build a recommendation engine for the e-commerce industry. A recommendation engine is an algorithm that analyses the user behavior (review ratings in this case) using various data analysis techniques to figure out the items that match the users' preferences and purchase patterns.

The aim of any recommendation engine is to stimulate demand and engage users. Recommender systems have become increasingly popular nowadays and mostly used in digital domains like e-commerce, social and media, healthcare, entertainment, hospitality and travel industries.

Recommendation engine has multiple benefits and popular benefits are described below–

* Increase average order value
* Driving up-sell and cross-sell
* Increase number of items per order
* Increase in shoppers to customers conversion rate
* Increase click-through rate (CTRs)
* Increase customer satisfaction and retention

The need of our capstone project is to do exploratory data analysis and understand the functioning of various algorithms and develop a product recommendation engine that can be further used to simulate recommendations across industries.

1. **Scope, Problem statement & objectives** 
   1. **Scope:**

Building recommendation models (Popularity based and User-based collaborative filtering etc.) using Amazon product review and metadata for the “Grocery and Gourmet” category, which would predict or help users in choosing items that are similar to other items they like.

The information about users and items is stored in a matrix that is modeled and used to make predictions i.e. the recommendations.

* 1. **Problem Statement:**

We would like to recommend the list of products which a customer is likely to buy based on customers review.

* 1. **Project Objective:**

Building a recommendation engine for the e-commerce industry with the goal to recommend new products to each user based on reviews and ratings given by customers.

1. **Data Sources and Description**
   1. **Data Source:**

The source of the datasets can be traced here -

1. <http://deepyeti.ucsd.edu/jianmo/amazon/index.html>.
2. <https://nijianmo.github.io/amazon/index.html>
3. deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Grocery\_and\_Gourmet\_Food\_5.json.gz
4. deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Grocery\_and\_Gourmet\_Food.csv

The data set we used is Amazon review and metadata for the “Grocery and Gourmet” category from May 2000 to October 2018. The dataset size for the Grocery and Gourmet food category is ~ 3.2 GB (*Review and Metadata dataset present in JSON file format*). and contains the variables as mentioned below: -

* 1. **Data Description** 
     1. **Review Dataset**

**Variables:**

1. reviewerID - ID of the reviewer, e.g. A1QVBUH9E1V6I8

2. asin - ID of the product, e.g. 4639725183 (Foreign Key)

3. reviewerName - Jamshed Mathur

4. reviewText - text of the review

5. Rating- 1-5 integer rating of the product which user has given for the product

6. summary - summary of the review

7. Review\_Time - time of the review

8. Unix reviewTime - time of the review (raw)

9. Product title

10. Product description

* + 1. **Metadata Set:**

1. Category

2. Description

3. Title

4. Brand- text of the review

5. Rank

6. Also\_view

7. Price

8. Asin (Primary Key)

9. Also\_buy

10. Image

11. Date

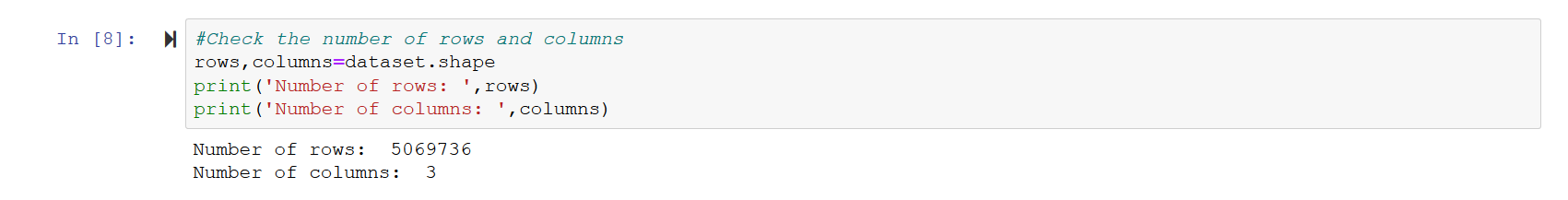
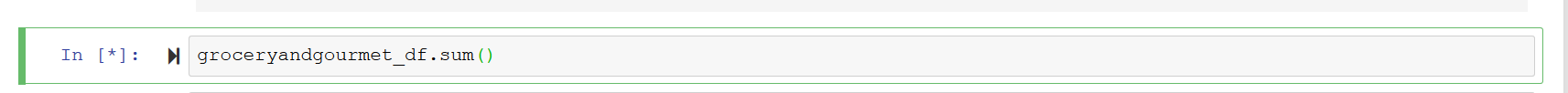
12. Feature

* + 1. **Volume of the Datasets:**

Grocery and Gourmet Food (3.20GB)

1. Grocery and Gourmet review dataset (519 MB:- 1143860 rows, 29 columns)

2. Grocery and Gourmet metadata (2.68 GB:- 287209 rows, 66 variables)

* + 1. **No of Rows and Columns of Review Dataset**
    2. **Missing Data Information**

Reviewer\_ID :- 0

Asin:- 0

Reviewer\_Name :- 0

Review\_Text :- 0

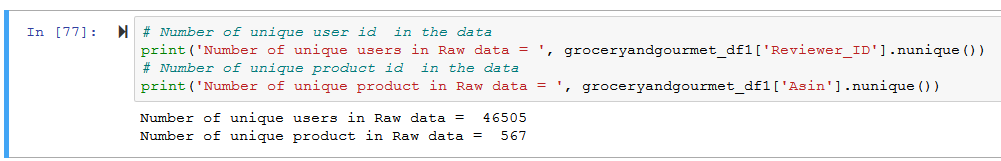
Rating :- 0

Summary :- 0

Unix\_Review\_Time :- 0

Review\_Time :- 0

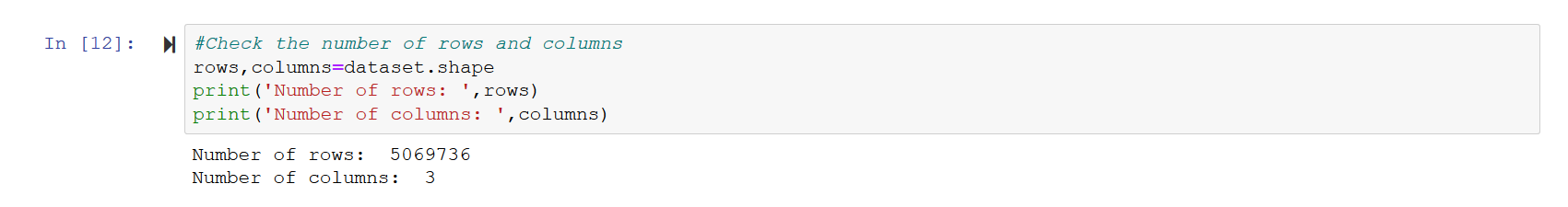
* + 1. **No of Unique Users**

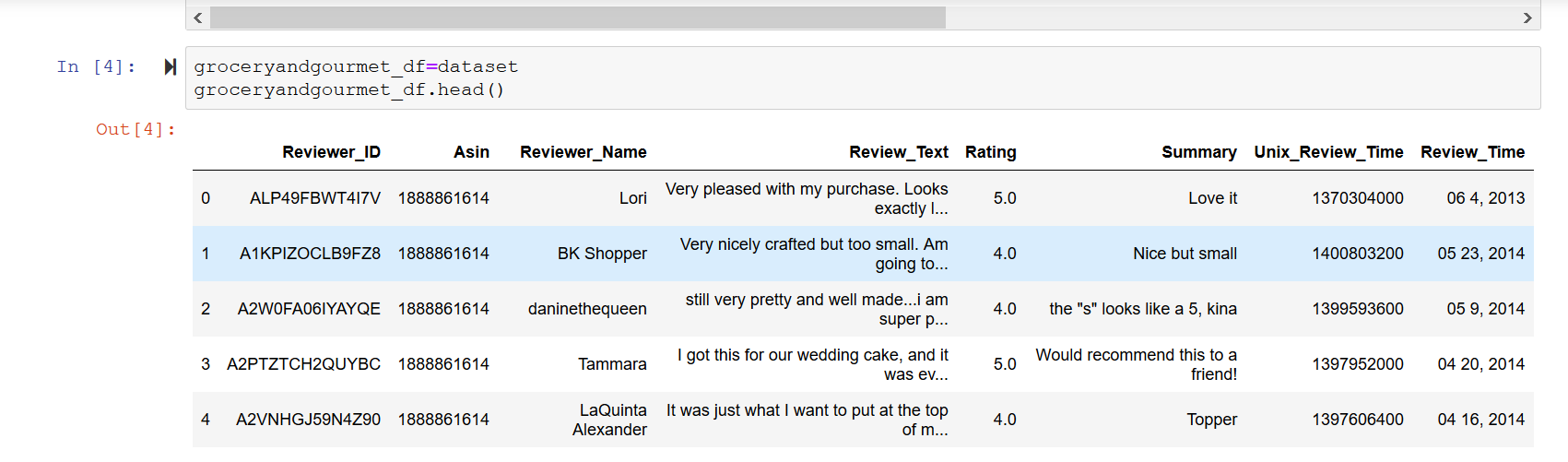


1. **Analytical Approach**
   1. **Basic Data Analysis and Exploratory Data Analysis (EDA)**
      1. Import all the required packages for exploring the data structure and performing various statistical analyses.

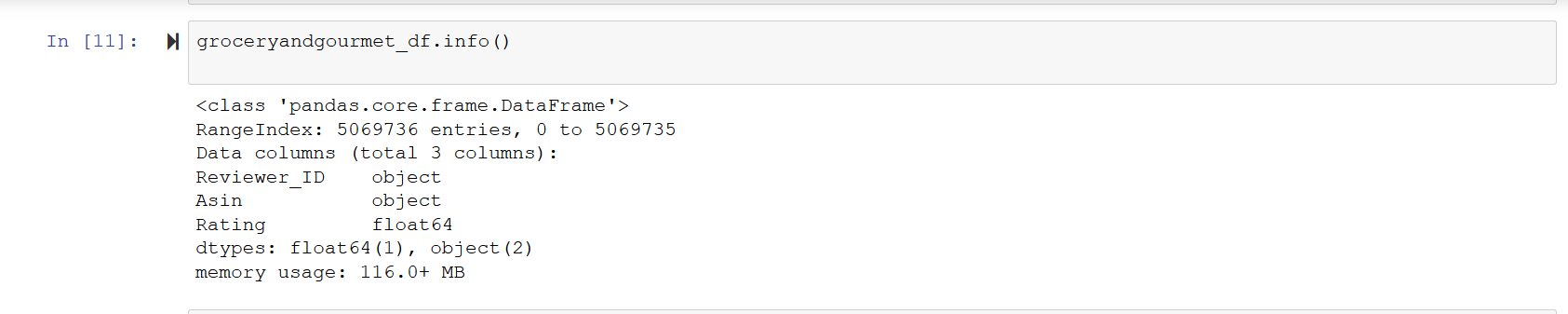


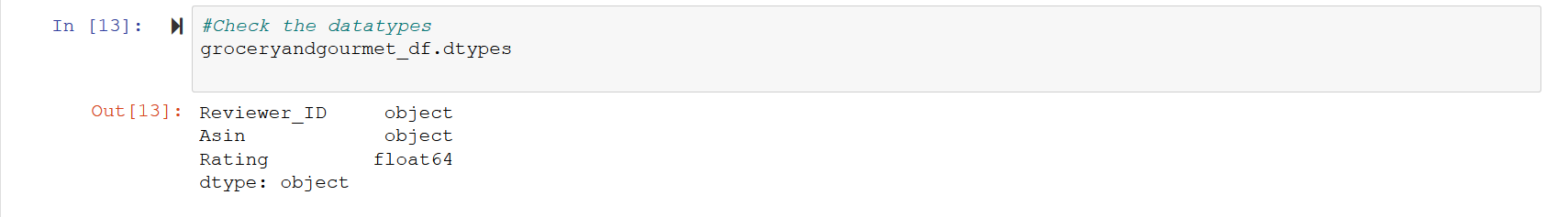
* + 1. Data set Overview:





* + 1. Variable information and Data types:

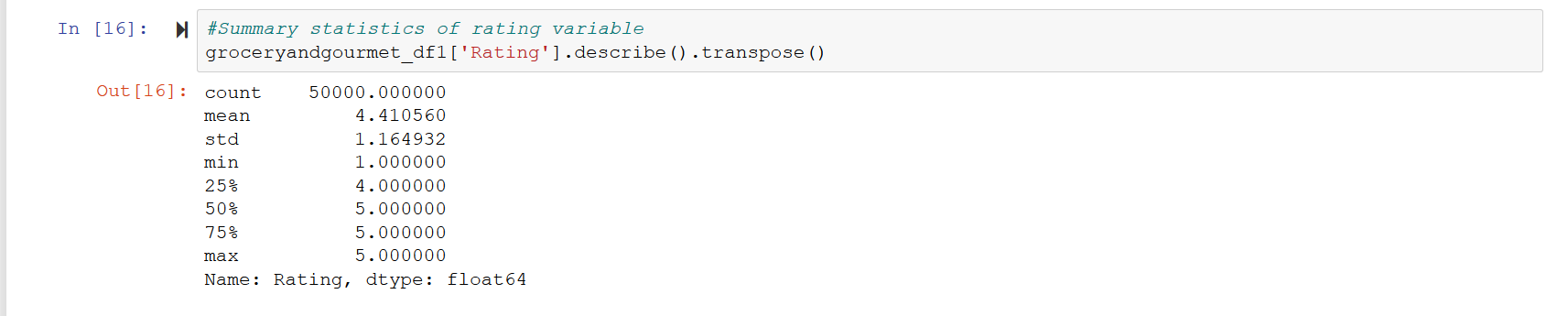


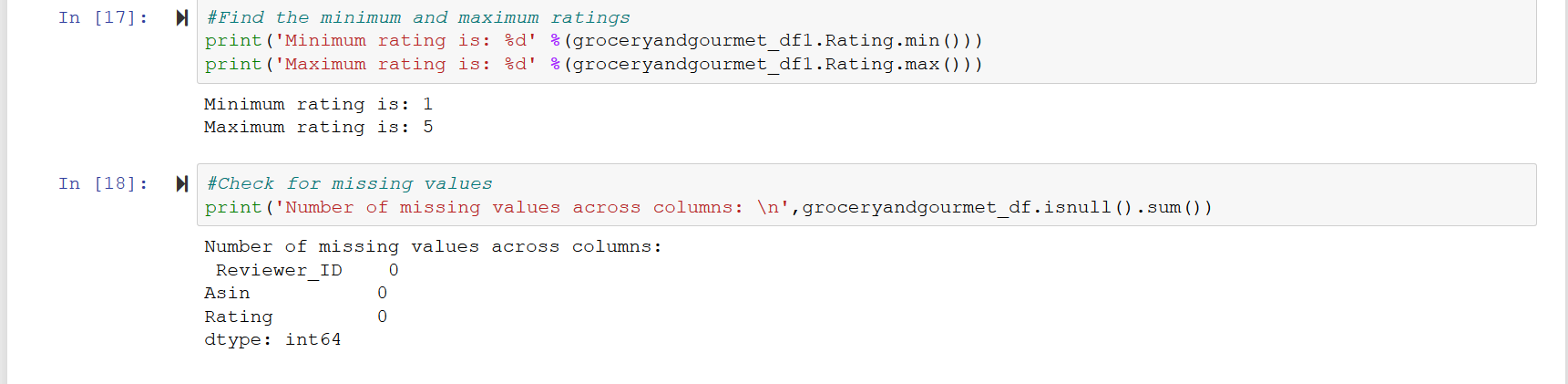


* + 1. Response variable: Rating (The range is from 1-5, 1 being the lowest and 5 being the highest, it is the rating that the user has rated a product)

### **5-point summary**

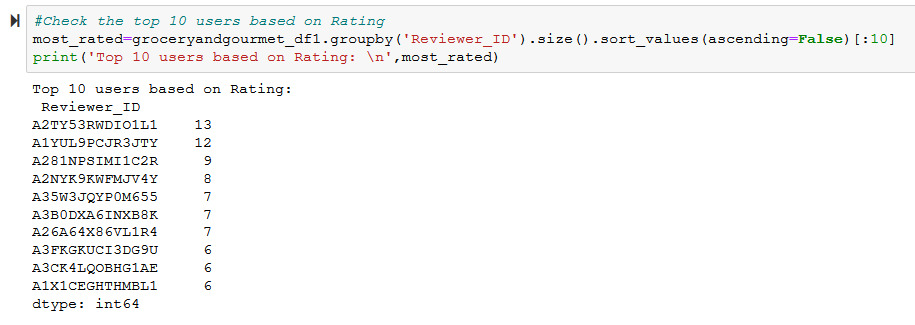
### The variable which is taken into consideration: Rating





### **Customer Profiling**

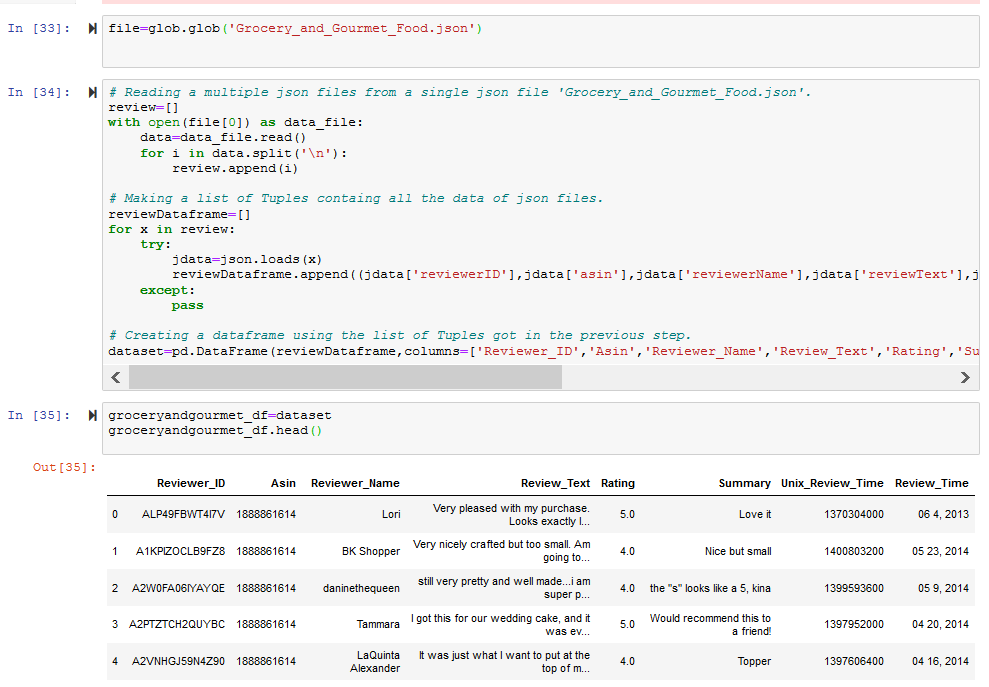
### Exploratory data analysis of top users who contributed to reviews and top items rating patterns.



1. **Data Pre-processing**
   1. **Data Pre-processing**

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Below are the steps that we have used to read and pre-process/clean the data:

### **Reading the review dataset:**



* + 1. **Cleaning the dataset:**

Cleaning (Data Processing) was performed on 'Grocery\_and\_Gourmet\_Food.json' file and importing the data as pandas DataFrame.

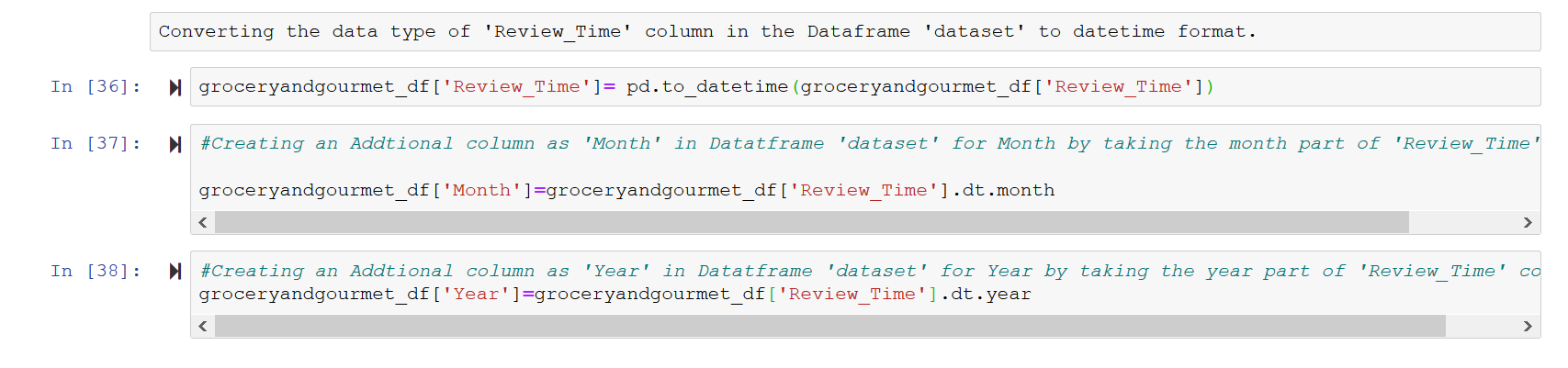
### **Tackling the Review\_Time variable**

1. We converted the data type of the “Review\_Time” in the data frame dataset to datetime format. As there was no direct function to convert the “Review\_Time” variable to a datetime format, we segregated each entity i.e. date, time & month separately and then merged into a new format.

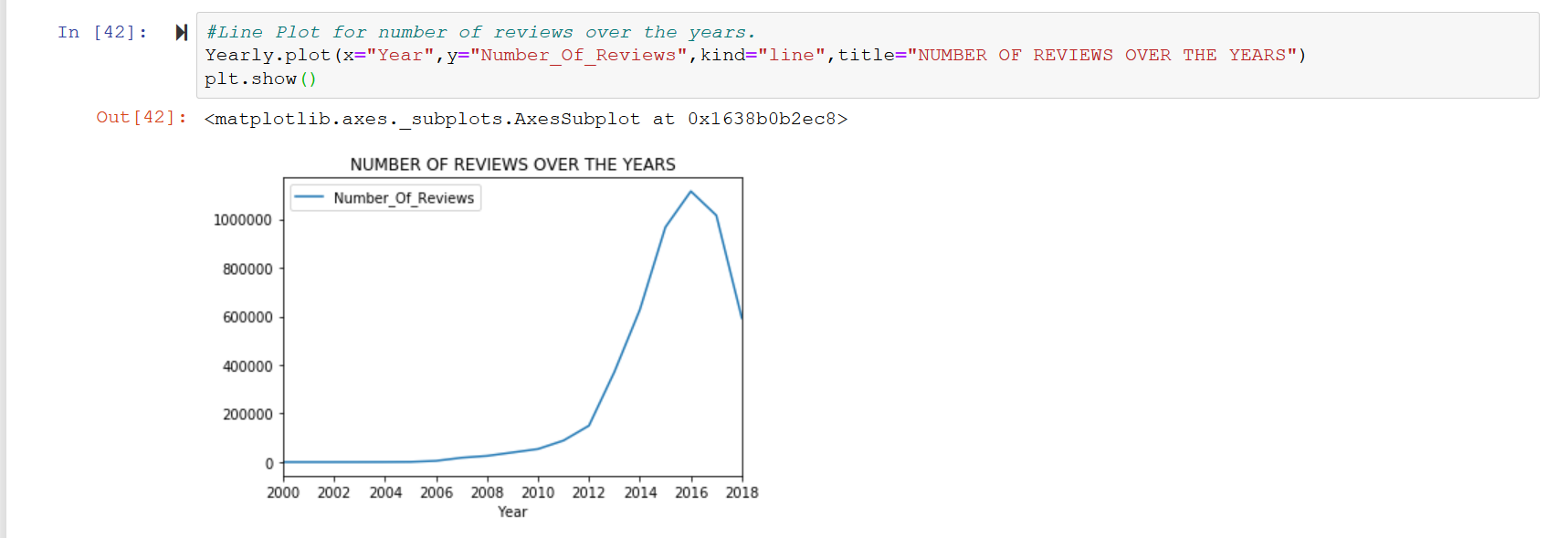
e.g. Format in dataset – (11,19,2014) is converted to (2014-11-19)

1. We have used (str\_trim) function from (stringr) library to trim the white spaces and then saved the year, month and date in new variable “new date”.
2. This variable “new date” is then merged with existing dataframe.
   * + 1. **Removing the unnecessary columns**
3. Since we had unnecessary columns that we are not using for the analysis of the dataset, hence we have removed columns 11 to 29.
4. After removing the unnecessary columns, the new data frame has 14 columns which will be used for Exploratory Data Analysis (EDA).
   1. **Feature engineering:**

* Converting the data type of 'Review\_Time' column in the Dataframe 'dataset' to datetime format.
* Creating an additional column as 'Month' in Datatframe 'dataset' for Month by taking the month part of 'Review\_Time' column.
* Creating an additional column as 'Year' in Datatframe 'dataset' for Year by taking the year part of 'Review\_Time' column.

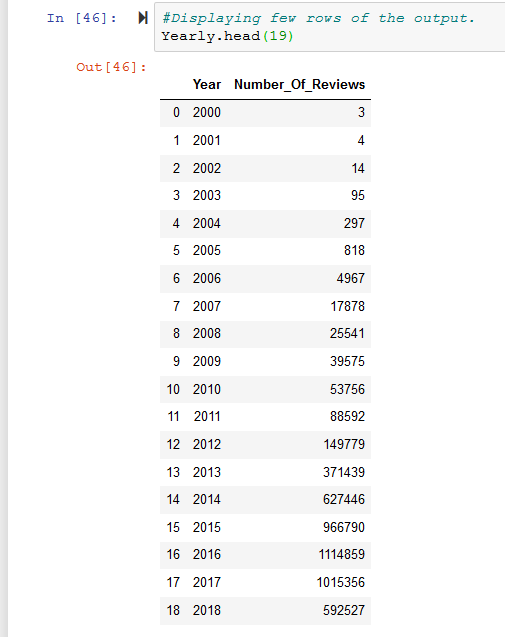
1. **Exploratory Data Analysis**
   1. **Number of reviews over the years**

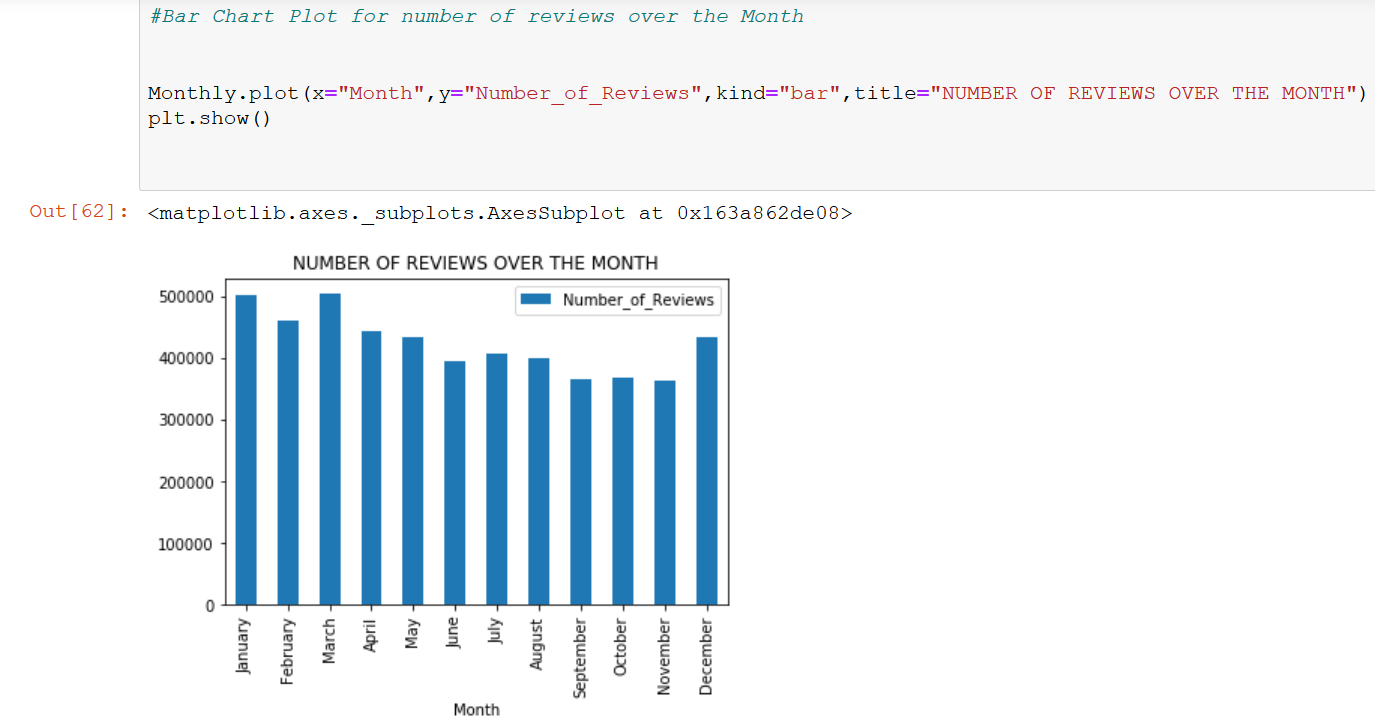
* Grouping by year and taking the count of reviews for each year.
* Below is the Line Plot for number of reviews over the years:



*Figure 1*

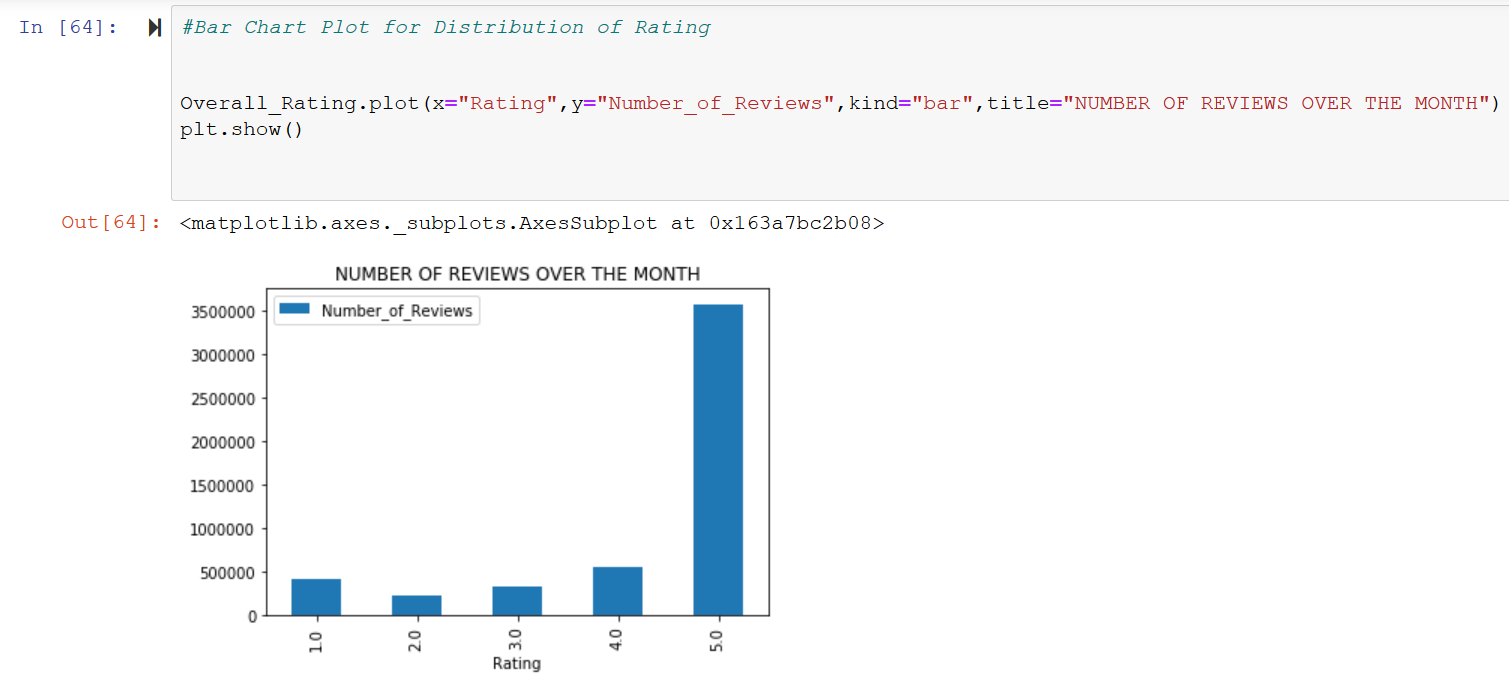
* Insights from the below Line Plot for the number of reviews over the years: As we can see from the above graph that the number of reviews has been constant from 2000 till 2008 and then it has gradually increased from 2008 till 2012. There is an exponential increase in the review from 2012 till 2016 and then it has decreased in the last two years. Hence, we can say that the review count has increased over the years.
* The number of reviews in the year 2000 to 2018 is shown below:



* 1. **Number of reviews by month over the years (2000-2018)**
* Grouping on Month and getting the count.
* Replacing digits of the 'Month' column in 'Monthly' data frame with words using the 'Calendar' library.
* Insights from the below Line Plot for the number of reviews over the years: Users have given more reviews for the product in the month of January and March.

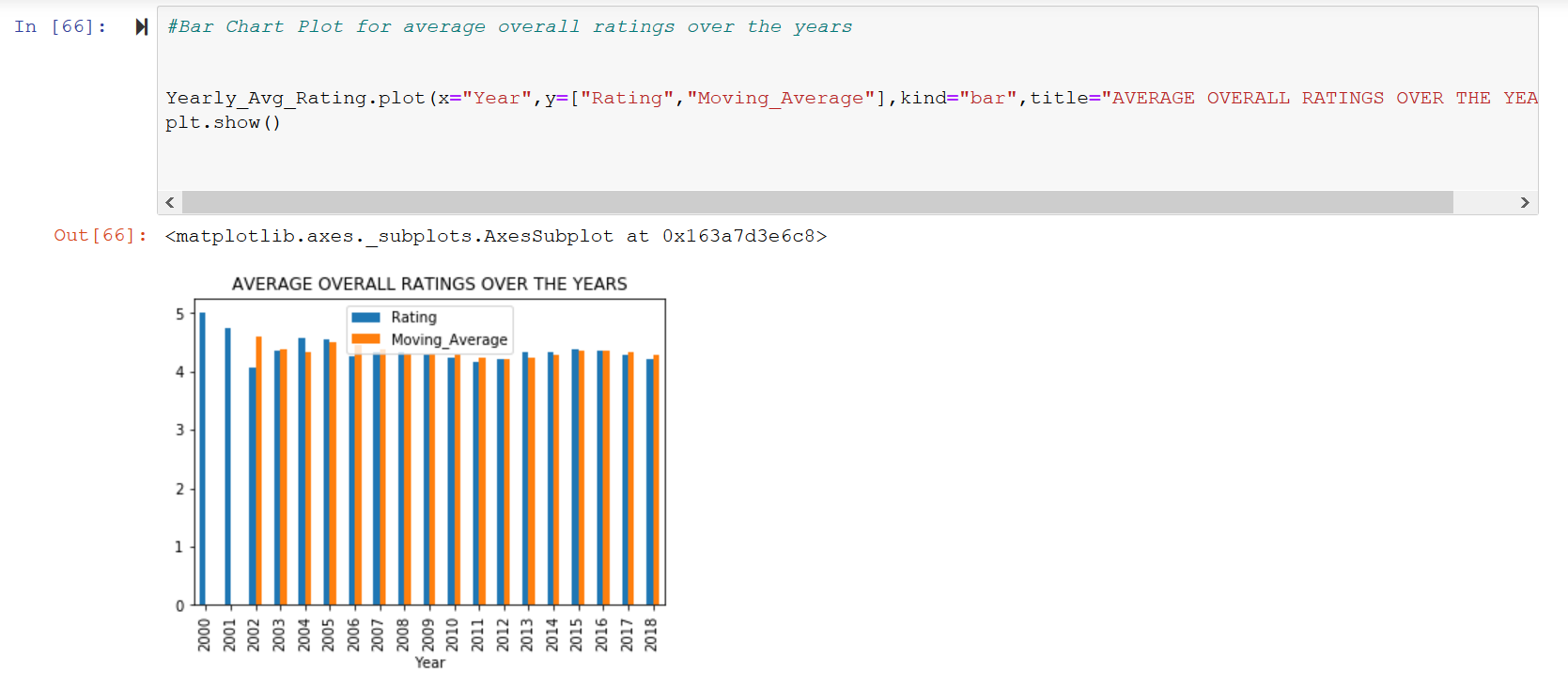
*Figure 2*

* 1. **Distribution of overall rating**
* Grouping on 'Rating' and getting the count.
* Bar-chart plot for Distribution of Rating.Insights from the Bar Chart plot below: Out of 5 million users, 3.5 million people have given 5 ratings during the year 2000 to 2018.



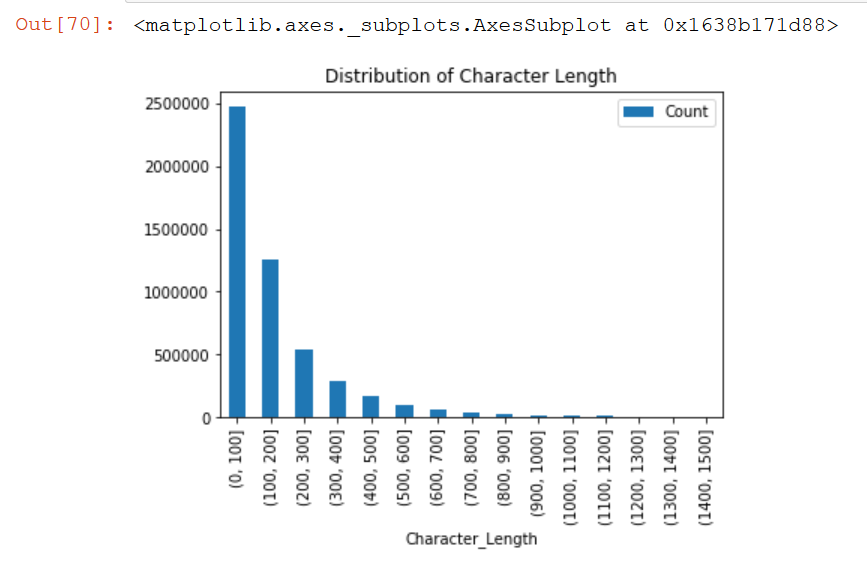
*Figure 3*

* 1. **Average overall ratings over the years**
* Grouping on Year and getting the mean.
* Calculating the Moving Average ith window of '3' to confirm the trend
* Insights from the below Bar Chart Plot for the Average overall ratings over the years: The average overall ratings from the year 2000 to 2018 has been always greater than 4.0



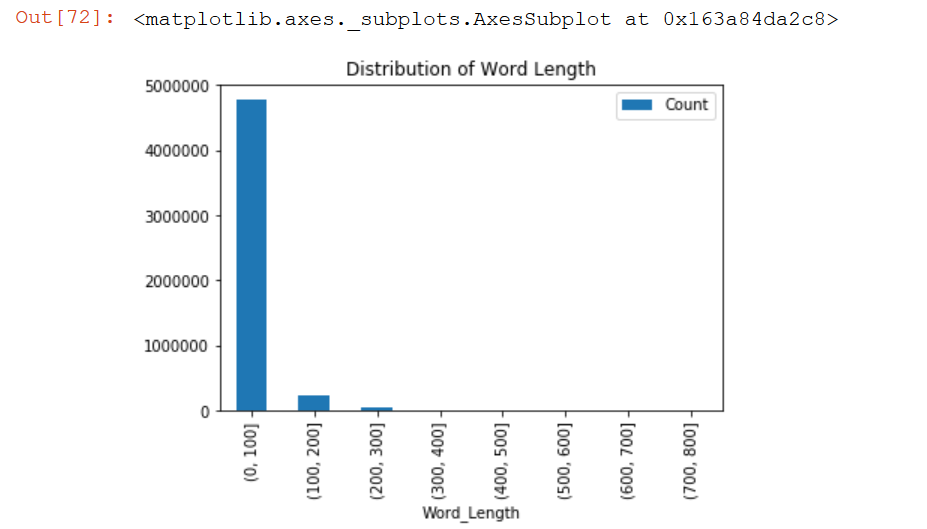
*Figure 4*

* 1. **Distribution of length of reviews on Amazon**
* Creating a new Data frame with 'Reviewer\_ID','Reviewer\_Name' and 'Review\_Text' columns.
* Counting the number of words using 'len(x.split())'
* Counting the number of characters 'len(x)'
* Creating an Interval of 100 for Characters and Words Length Value.
* Insights from the below Bar Plot for distribution of Character Length of reviews on Amazon:



*Figure 5*

* 1. **Distribution of Word Length of reviews on Amazon**

Creating a new Data frame with 'Reviewer\_ID','Reviewer\_Name' and 'Review\_Text' columns.

*Figure 6*

1. **Modeling approach**
   1. **Popularity-Based Collaborative Filtering**

Collaborative filtering is a branch of recommendation that takes account of the information about different users. Given a new user, the algorithm considers the user’s purchases and recommends similar items.

As the name suggests the Popularity based recommendation system works with the trend. It basically uses the items which are in trend right now. For example, if any product which is usually bought by every new user then there are chances that it may suggest that item to the user who just signed up.

There are some problems as well with the popularity-based recommender system and it also solves some of the problems with it as well. The problem with a popularity-based recommendation system is that personalization is not available with this method i.e. even though you know the behavior of the user you cannot recommend items accordingly.

We took 50,000 observations for building the user-based collaborative recommendation system.

* + 1. **Building the Popularity Collaborative Filtering model**

We’ll use UBCF, singular value decomposition method, cosine similarity model our training data. The data is automatically normalized to control for rating bias.

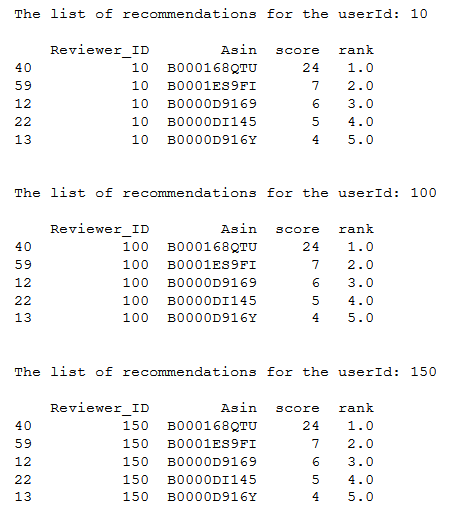
We took 50,000 observations for building the user-based collaborative recommendation system.

* + 1. **Train Test Split**

The data was further split into 70:30 into the train (train\_data) and test dataset (test\_data) respectively.

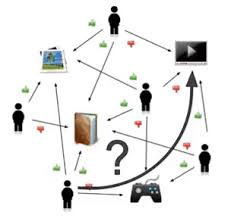
* + 1. **Predictions on the test set**

The prediction was populated for user id 10, 100 and 150 respectively.

****

* 1. **User-based collaborative filtering**

The User-based collaborative filtering recommends items based on whether similar users purchased or rated them similarly.



For each user, these steps are taken:

1. Measure how similar each user is to other users. As with IBCF, popular similarity measures are Pearson correlation and cosine similarity.
2. Identify the most similar users. Choices can be based on similarity to the top k users (k-nearest\_neighbors) or users above a defined similarity threshold.
3. Rate the items purchased by the most similar users, either by averaging or weighting the nearest users.
4. Pick the top-rated items.
   * 1. **Building the User-Based Collaborative Filtering model**

We’ll use UBCF, singular value decomposition method, cosine similarity model our training data. The data is automatically normalized to control for rating bias.

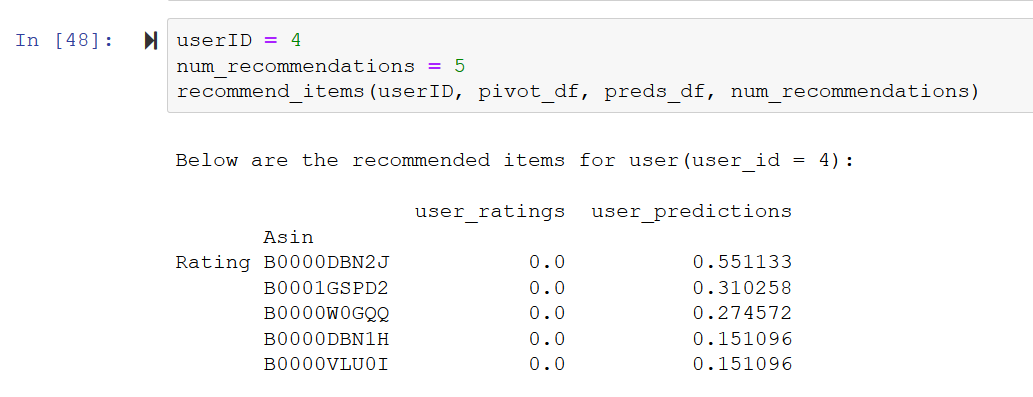
We took 50,000 observations for building the user-based collaborative recommendation system.

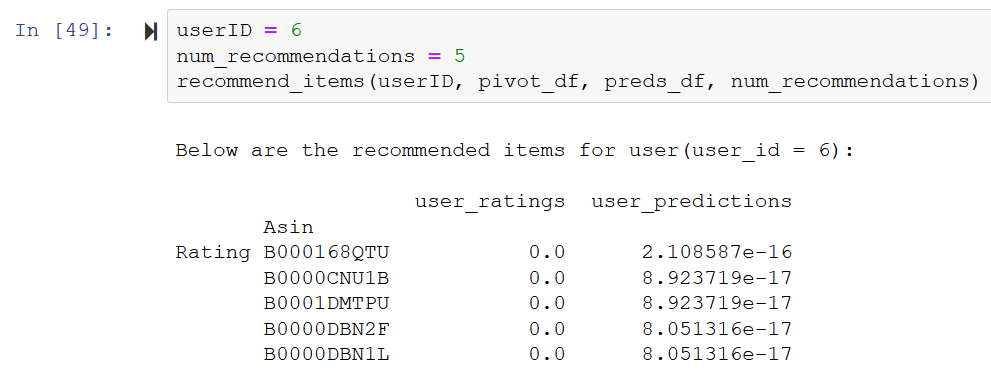
* + 1. **Train Test Split**

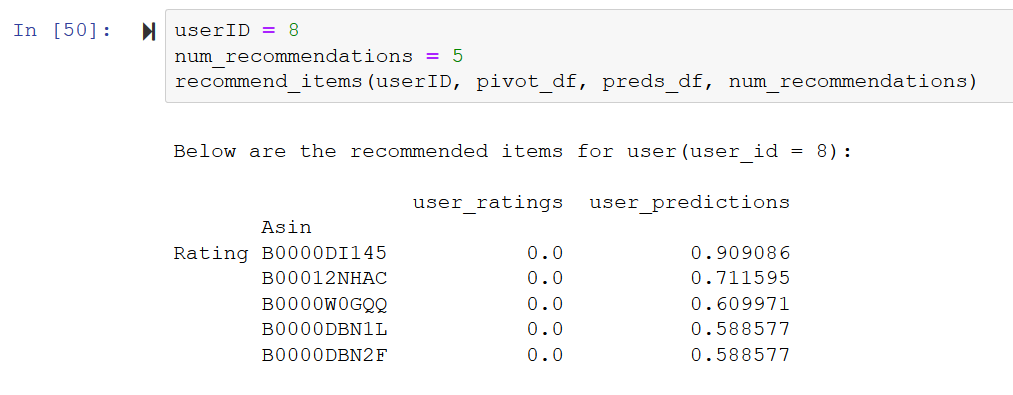
The data was further split into 70:30 into the train (train\_data) and test dataset (test\_data) respectively.

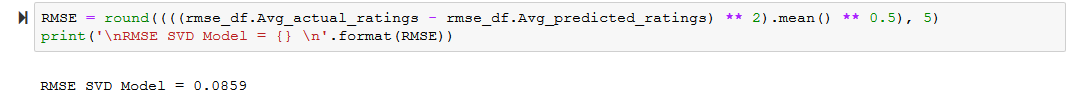
* + 1. **Predictions on the test set**

As with IBCF, we use the model to predict a Top 5 list for users (User id 4,6 and 8).







* + 1. **Evaluation of Collaborative Filtering Technique**

1. **Insights and Findings from Exploratory Data Analysis**

* There has been exponential growth for Amazon in terms of reviews. This can be inferred that probably sales also increased exponentially during this period. From 2016 until 2018 there is an observed dip.
* Buyers generally give more reviews from December to March.
* More than 80% of the reviews give a 4 or 5 star rating, with very few giving 1, 2 or 3 stars relatively.
* Average Rating over every year for Amazon has been above 4 and also the moving average confirms the trend.
* The majority of reviews on Amazon have a length of 100-200 characters or 0-100 words.

1. **Future Course of Action**

* We will expand our use cases to build a recommendation engine using various techniques based on both users (new and repeat) and product reviews (positive and negative). Recommended techniques to be built are Content-based, Item-based collaborative techniques based on user's choice and characteristics.
* Also, we will be looking at the market basket analysis to find out what the name and types of products are usually bought together.
* We will be applying the recommendation engine on different users and observe the recommended outcome. We will also link with the metadata to populate the item name and the user name.

1. **References and Bibliography**
2. Justifying recommendations using distantly-labeled reviews and fined-grained aspects by Jianmo Ni, Jiacheng Li, Julian McAuley (***Empirical Methods in Natural Language Processing (EMNLP)*, 2019**)
3. <http://deepyeti.ucsd.edu/jianmo/amazon/index.html>.
4. <https://nijianmo.github.io/amazon/index.html>
5. deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Grocery\_and\_Gourmet\_Food\_5.json.gz
6. deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Grocery\_and\_Gourmet\_Food.csv

This dataset was shared purely for research purposes.

**9. Appendix**

Python code:

