Big Data (6CS030)

Anime Recommendation System

Student Number	: 2049855
Full Name	: Jenny Rajak
Group	: L6CG1
Module Leader	: Mr. Jnaneshwar Bohara
Lecturer	: Mr. Jnaneshwar Bohara
Submission	: 02/05/2022

Table of Contents

Table of Figures

1. Introduction

1.1. Big data (BD)

Big data is a massive stack of data numbers. It keeps on adding up with time, which is why due to its enormous amount, data managed traditionally fails to handle such data, and big data came into the picture with the capacity to hold data that are big in numbers. Some of the examples of BD is Google, social Medias. (Taylor, 2022)

Characteristics of big data:

Volume: Data are generated daily increasing in size. The need to in increase the storage capacity is also added on. In short, it is the large amount generated data.

Variety: It refers to the uncertainties of the data.

Velocity: Whereas, it is associated with the rate of the produced new data and speed of the data. (Siddharth_s, 2020)

1.2. Big Data Process

The big data process refers to the working of the big data that includes four sub-task, gathering data, and clarifying data, tidying up the data, and observing the data.

Gathering data

Data gathering is done by both big and small companies. Those data are both found structured and unstructured, gathered from various sources. The storage of such data is necessary; to store such data, companies use data warehouse.

Clarifying data

After the gathering and conserving of data is completed it needs to be managed properly in order to get exact output. The need to manage data is especially needed when the data is huge and scattered.

Purifying data

An enormous amount of data can contain duplicate, null, garbage values which might result in false observations. This is the reason why it is encouraged to format data correctly to receive an exact result.

Observing data

The humongous amount of data needs time to be formatted into a piece of useful information. Then, those useful data can help in concluding huge perceptions. Additionally, those data can be used in the fields of deep learning, data mining, and predictive analytics. (Tableau, 2022)

1.3. Sentiment Analysis

It is the method of recognizing sentiments through an online patch of writings. Emotions can be denial, neutral, or positive. There are tools for analyzing feelings online on author writing as negative, neutral, or positive. (Tom, 2021)

1.4. NOSQL

It is non-relational database. It is popular for storing unstructured data. NOSQL is quite popular for handling data that are big in size, also which demands for huge storage. Today, NOSQL is used everywhere due to its scalability. For instance, real-time Medias and big data use NOOSQL. Examples: Facebook, twitter. (Taylor, 2022)

1.5. Search Engine Technology

Search engine technology is the machinery that helps you search things through your small piece of writing and provides the exact or similar kinds of suggestions. One of the most promoted search engines is Google. (Knauff, 2018)

1.6. Data Warehousing

The DWH is the repository that stores data from a system that is operating, and those data are made available for day-to-day operations. Data warehousing electronically stores data and is different from another operational environment. Here, data are clustered together, and files are easily available for performing daily tasks. DWH should be reinforced in the business nowadays. (Ahmed, 2020)

1.7. Data Mining

They are the knowledge detection in data also known as KDD. It is used in converting the raw data into useful data that provides valuable information. They are also efficient for uncovering useful information from a large dataset. Since data keeps growing with time, implementation of data mining is a game-changer as converting raw data into useful insights is time-consuming. Data mining is divided into two main tasks: they can predict output through machine learning algorithms or can describe the targeted dataset. (Education, 2021)

1.8. Hadoop

It is the framework created by java to handle big data. It uses parallel processing and distributed storage to take care of enormous data. Hadoop is used frequently by the people who analyze data. It consists of three main components they are:

HDFS: It is the unit that is responsible for storing.

MapReduce: It is the unit for processing.

YARN: It is a unit that manages all the resources that are required. (Zhasa, 2021)

1.9. MongoDB

It is an open-source NoSQL database that is document-oriented. It is easy to use scalable and provides high performance. It performs tasks on the collection of documents suitable mostly for JSON files. (tutorialspoint, 2022)

2. Anime Recommendation System

Anime business are developing quickly with new works of anime being added daily, it is quite popular among the young generations, making it challenging for anime lovers to search for anime that match their interests. Therefore, the need for building a recommender system engine comes in handy for users to search for similar kinds of anime that they prefer to watch.

3. Abstract

3.1. Generic Information

The project is about developing a recommendation system that recommends anime by implementing big data.

3.2. Problem Statement

Webpages that use recommendation systems are less in number due to which the websites without recommendation systems are losing their users.

3.3. Aims & Objectives

The idea behind developing such a system is to make a recommendation system that recommends popular anime. While developing such a system, we will learn about various approaches needed to build a recommendation system.

3.4. Contribution of the work

During the time of working on this project, daily research of datasets, recommendation system approaches, MongoDB queries, and similar systems contributed to finishing up the project.

3.5. Social/Research community impact of the work

The impact of the work was truly worth it in the deadline as we were aware able to produce a task that recommends popular anime, we got to research big data, a recommendation system, and many more. In short, in the end, the project was quite knowledgeable.

4. Background of the study

4.1. Generic Information

As mentioned earlier, it is a project built to recommend anime on the basis of the users' inputted ratings. A recommendation system engine is made successfully that recommends similar types of anime to the user using two types of method cosine similarity and collaborative filtering. The coding is done in python where necessary data preprocessing is done. In addition to that, since it is a big data module the implementation of NOSQL is also necessary so mongodb and python was connected using pymongo and queries were performed.

4.2. Problem Statement

Anime is a popular Japanese cartoon, famous among both adults and children. Due to its popularity, anime works have been developing drastically, resulting in users facing difficulty to search for anime to watch that matches their preferences. Also, the web pages that use recommendations to recommend anime to the users are rare affecting the loss of website users. This problem can easily be tackled by introducing a recommender system to anime websites.

4.3. Aims & Objectives

Aims:

Successfully develop a recommender engine that recommends popular anime to the users so that they do not have to search for the anime works manually.

Objectives:

- To build a personalized recommender system that recommends anime with specific users' interests.
- To use the past recorded rating of users to determine the future recommendation of anime for that specific user.

- To benefit the anime sites with a recommender system for increasing their number of users.

4.4. Organization of the Report

We have structured the report according to the description given in the coursework file.

Introduction:

This part contains an introduction to big data and big data-related terms.

Title:

Title of the project and a short description of the project work.

Abstract:

This section is a little bit extended version of the project.

Background:

This part consists of a detailed explanation of the project.

Related work:

Here, we have mentioned the research papers we have found and summarized them.

Methodology:

We have discussed all the methods we have set up to complete this project.

Result and discussion:

Here, all the result and explanation of it with evidence is provided.

Conclusion:

We have mentioned our learning, outcomes, and findings.

References:

We have cited the works of the author, which we have mentioned in our report.

5. Related Work

5.1. Big Data Analytics for Search Engine Optimization

Big data analytics plays a critical role in optimizing the search engine traffic in the WWW circle. Untamed huge information with enormous volume also, speed are created in regards to the cooperation's of user with search engine and sites and how they individually respond to the list items and the substance they get. In the time of big data, customized content that reflects quality recommends users with data of their preferences on the website, resulting in their ease of use. Nonetheless, regardless of the large availability of data and their utilization in the web frameworks, some areas still need improvements. One of the main reasons is data overloading and, consequently, facing trouble in data visualization, preprocessing, and insights of the outcomes. This paper discusses the approaches for implementing big data analytics efficiently for increasing website performance and their contributions to SEO goals. Here, the methodology is applied in the cultural heritage domain.

A cultural heritage domain is a website built to preserve the ancestors' legacy in the society. And also to spread awareness about the cultural importance. This website portrays the local cultures, and traditions, in an attractive way educating their interested end-users locally and globally. Though managing the website is difficult as this web consists of a large number of datasets, links, and images that necessarily needs to be optimized by utilizing big data analysis, which can benefit the website with good cultural content in accord to user preferences. The methodology divides into three hypothesis which are:

Assumption (A1): Size of a website leading to lower and higher search engine traffic.

A news question is raised on the size of the website leading to higher search engine traffic, which is related to the compatibility of SEO and if the CHI website follows it. Big data analytics offers a new platform for SEO deployment and planning. The plan is to collect data through preprocessing and API (application program interface) integration which will result in the plurality of behavioral and technical SEO variables impacting the

CHI website's organic search engine traffic variance. Also, based on PCA, the researcher divided the behavioral and technical variables into four factors: User behavior, Website security, SEO Crawling, and Website loading speed.

Assumption (A2): CHI (Cultural heritage institution) organic search engine traffic percentage is increased due to the impact of the SEO crawling factor.

Loading time affects the end-users. If the loading time is less, the user feels at ease using the website, and their interaction and engagement increase. Example: BBC lost 10% of their visitors for consuming time to load their content. Higher loading time negatively impacts user experience leading to low engagement and greater bounce chances. The critical issue is connected with redirections of web pages inside and outside of the website and pressure and modification of CSS and JavaScript. Thus, the proper examination should be done to find out the compatibility of a website's loading time. A higher percentage loading speed is directly proportional to the percentage of increased organic search engines.

Assumption (A3): CHI percentage of organic search engine traffic is caused by the website loading speed.

There is no prior evidence that the website security has impacts in the organic search engines. Nevertheless, security plays a vital role in the SEO context.

Assumption (A4): CHI security factor has impacts on increase percentage of Organic search engine traffic.

If a user spends much time interacting with the website, exploring its content the probability of content increases for the potential users. In the world of, big data there is a need of optimization and personalization of data by the people and for the people.

Assumption (A5): CHI user behavior impacts on increase percentage of organic search engine traffic.

On the basis of the assumption made three methods were introduced, which has the capacity to tackle several issues that arises during big data analysis. They are as follows:

- With the organizations' defined key performance indicators alignment, summarization and validation of retrieved big data analytics.
- The advancement of indicative exploratory models that determine the circumstances and logical results connections between the measurements.
- The prescient model turn of events and the course of reproduction for improvement purposes.

  Moreover, the output emphasizes to the reclassification of the SEO topic. We can conclude that if the compatibility of SEO factors are higher, the ranking of search engine visit percentage is also higher. Notwithstanding, the primary point of web crawlers is affirmation that they give the most subjective substance, in the most elevated volume, in the quickest time to their clients, as per their preference. In this regard, Web engineers and content makers ought to have reasonable measured markers to assess and advance their site execution and content. (Drivas, et al., 2020)

5.2. Collaborative Recommendation System in Users of Anime Films

With technology being developed rapidly, most of the tasks can be performed digitally. A few data on the internet increasingly more with the development of the web progressively which records each data got. As an outcome, much data is questionable. It makes users difficult to come by their inclination in light of the fact that the data can be unessential to them. To solve this problem recommendation system has been developed.

The recommendation system is beneficial to users as it helps to find users and search for data that match their preferences without searching it manually. In the field of manga and anime, new works have been introduced daily with an interesting storylines. As for anime lovers, the growing manga and anime make it tough to let them choose which work to watch first. In this paper, researchers have discussed using collaborative filtering for recommendation systems. It is one of the most used methods for finding similarities between users, films, and shows to guess user preferences.

Related work

In this research, researchers use big data with map-reduce to develop a collaborative reconciliation system able to reduce the computational process cost. They also proposed a recommendation system to recommend films using collaborative filtering that concentrates on ratings provided by the users for providing recommendations to other users. In the system, the user can choose from the given attributes, and then the system provides the user with the list of recommended films based on cumulative weights of different using the k-means algorithm films based on the highest order. The reconciliation system uses both user-based filtering and collaborative.

Proposed Method

- Normalize the ratings of the movie using mean
- Randomize small values for movie feature vector X and parameter, 0
- Using gradient descent minimizes the cost function
- Movie rating prediction
- Recommending top movies by calculating feature difference

Result and Analysis

The dataset was taken from Kaggle namely anime data and ratings on anime. The collaborative filtering method works by guessing the rating of the user's data on the basis of the similar history of previous users. It is performed using matrix factorization and the training data recommendation system is built using ALS (Alternating least square).

Conclusion

A recommendation system is built using users' watch history and predicts whether the later users like the particular anime or not. Although, a 100k dataset was taken due to the limitation of not accepting the large datasets. It is said that if 1 million datasets are used for prediction it might provide much better results. (Girsang, et al., 2020)

5.3. An Intelligent Analytical Framework for Anime Recommendation and Personalization using AI and Big Data Analysis

The popularity of anime has increased within the past few decades resulting in the development of more new works. Anime is a Japanese cartoon watched by both adults and children. With the increase in anime works, users face difficulty in which anime to watch next. There is one simple solution for this problem which is to go to the internet and ask, yet the researcher wants to explore an easier way i.e. building a recommendation system that recommends anime that matches users' tastes. Here, the anime is recommended using AI. In this paper, users are using the data from MyAnimeList.

The anime enthusiastic faces a common problem when seeking suggestions for what to watch next, so they rely upon the forum for different users to suggest anime that might match their taste. One of those popular forums is Reddit. While observing the forum, it is found that when suggesting an anime to the newcomers' people argue with one another about the anime they suggested. Consequently, the newcomers fall into a dilemma on whom to listen to. Also, the process is time-consuming, which prevents people from watching anime. A similar recommendation system is found, that implements AI RikoNet. The method the investigators has used is SVM (non-linear support vector machine) for classifying vectors utilizing linear regression. SVM takes multiple values in consideration making the prediction more accurate, and the dataset used is from myanimelist.net well known largest dataset for ensuring accuracy.

They experimented to look after the effects on accuracy on the train data set size and genre used for SVM. First they changed the rows form 0—100 to 200-300 rows as training data, then second they modified the genre and observed, which genre impacted more on the SVM, for removing the redundant X variable and to find out the important variables. As the model becomes complicated if there is many variables of X, leading towards longer computational time.

There was some challenges faced on the current system which are as follows:

Finding useful information

Internet is loaded with useful information confusing people more about what they want. Similarly, the forum can help anime lovers seeking for which anime to watch but it is time consuming leading towards confusing the anime seekers more.

Save training time

There are few website that uses recommendation system one of which is RikoNet. It is very similar to what Netflix has. Here, users' data is recorded, then prediction is made and finally feedbacks of the users is taken in order to improve the site. The only problem is that it takes time and the users' feedback is needed constantly even though it is found to be effective in a long run. Additionally, it cannot be applied to small sites.

Choices are limited

Crunchyroll and Funimation has a developed recommendation system, big enough to hire professional people to recommend anime. Although, anime license is expensive, can cost millions of dollar. For instance, Evangelion is not recommended by them as it is licensed by Netflix. This sites only recommends works that are available with them.

Experiment

First experiment

100 rows of data were taken as the test size to check the test accuracy of how similar the model's prediction is compared to the validation set, but the validation set was not available. Because of this similarity between the whole data and the predicted data was checked. It was found that 100 rows were not enough to set off the previously mentioned issue. The desired output was sorting out the popular anime at the top and less known at the last. Rows were then taken randomly first from 0 to 100, then 100- 200, and lastly 200-300, to their surprise it was found that 100-200 and 200-300 yielded the same result so they decided on performing further research.

Second experiment

Since the theory is based on SVM operators. They initially stated that they will reduce the number of x variables for increasing the performance of the model. More normal classes would make support vectors that poke the relapse line in a more broad bearing in this manner covering more and prompting high precision. For the testing purpose the in the first iteration most common genre was taken: Drama, Action, Horror, and Comedy, then in the second iteration mix of both common is taken: Fantasy, Ecchi, Drama, and uncommon Game, Dementia and Demons, lastly, in third iteration uncommon tags is taken: Horror, Magic, Arts, Mecha, Military, Mystery, and Music. As for observing the result, the first iteration provided 48% accuracy and the rest of the three iterations did not reach half the accuracy. There was an unexpected result which was the uncommon tags scored the highest when mixed with the lowest. After the experiment, their hypothesis was proved partly true.

Conclusion

There was two experiment conducted from the first experiment it was found that using the top 100 anime acquired the lowest accuracy, the reason was due to the nature of SVM and popular anime squalls. And in the second experiment, it was found that uncommon genres yielded better accuracy. To conclude, to train the model 100-200 rows are considered, and working on all the genres delivered an excellent performance with an accuracy of 52%. However, there is a limitation of the model being built on a year-old dataset, which can cause issues regarding accuracy. (Ota, et al., 2021)

6. Methodology

6.1. Block Diagram

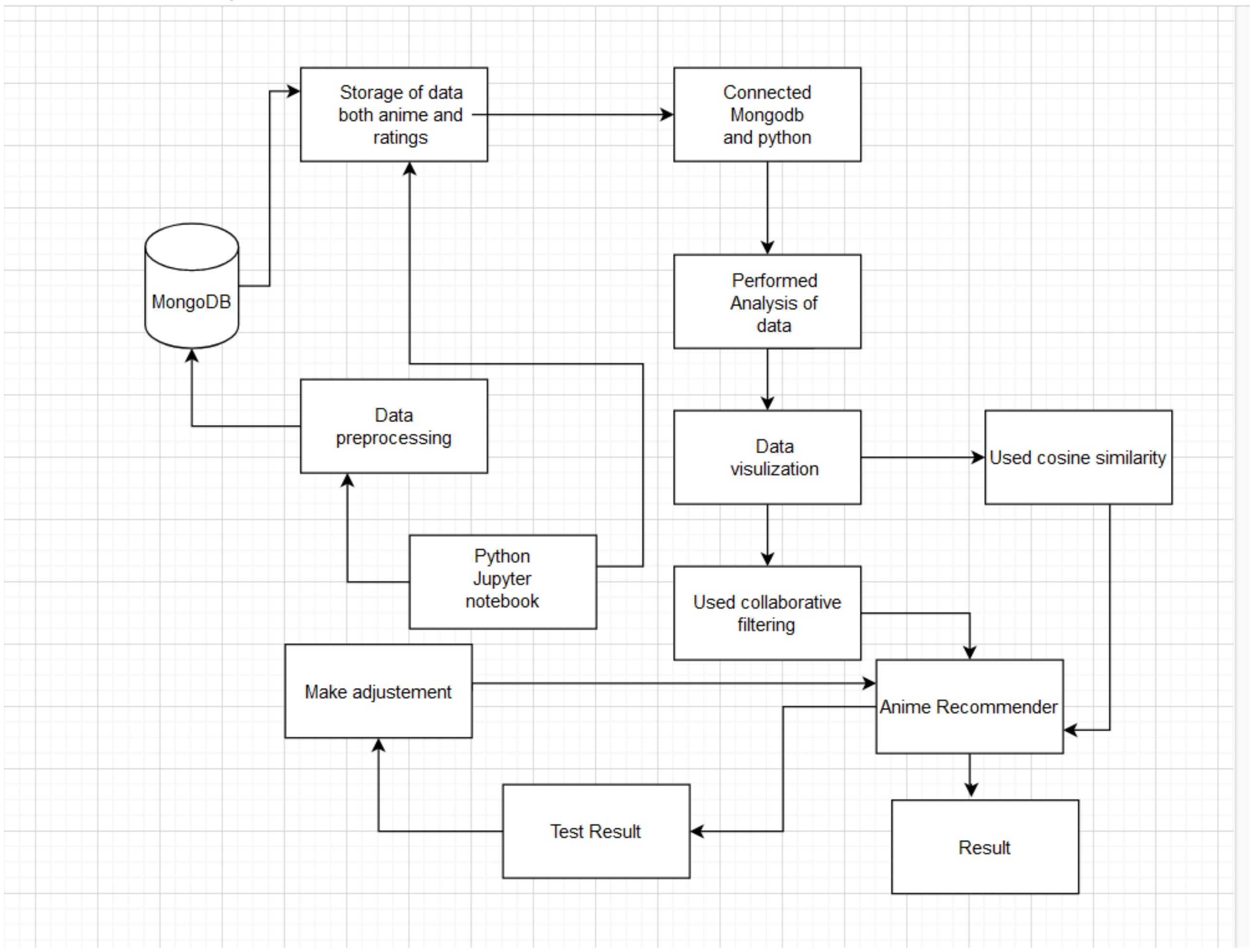


Figure 1: Block diagram for anime recommendation system

6.2. Data Reading and Exploring

Data reading is studying the meaning of each column, and going through all the rows. Data exploring is checking out, the shape, missing values, and data types of each column.

6.3. Data Analysis and Cleaning

Data analysis refers to the analyzing part of the data where we draw some insights about the feature and the targeted values. Data cleaning is the step where the missing values is replaced with some meaningful values, dropping the unnecessary columns that we think is not necessary for drawing conclusions, changing of necessary data types if necessary.

6.4. Collaborative Filtering

It recommends anime by comparing the users' given ratings and recommends other users with the highest rated popular anime. It was introduced, to overcome the drawbacks of content-based filtering, which recommends anime based on a popular genre. (Recommendation System, 2021)

6.5. Cosine Similarity

It is a metric used to compute the similarity scores between two anime and movie. It works by measuring a multiple-dimensional angle between two vectors. (Javed, 2020)

6.6. Data visualization

Data visualization is a pictorial representation of data which is frequently done to find out the relation between features to features, features to targeted variables, and also to find out their dependencies on one another.

7. Result and Discussion

7.1. Read and Exploration of Data

About Dataset

- This data set is taken from data. World and the main data source is MyAnimeList
- The dataset contains information on user preferences anime work from 73,516 users on 12,294 anime.
- Dataset consists of 12,294 rows and 7 columns
- Two datasets will be used for building the recommendation engine namely amine.csv(anime related data) and rating.csv(user preference related data)
- Features in anime.csv:

anime_id – unique id for identifying an anime

Name - full name of anime

Genre – genres of anime separated by comma

Type – TV, Original Video Animation, movie etc

Episodes – number of episode of the anime show (if movie 1 episode)

Rating – average rating of anime out of 10 for given anime

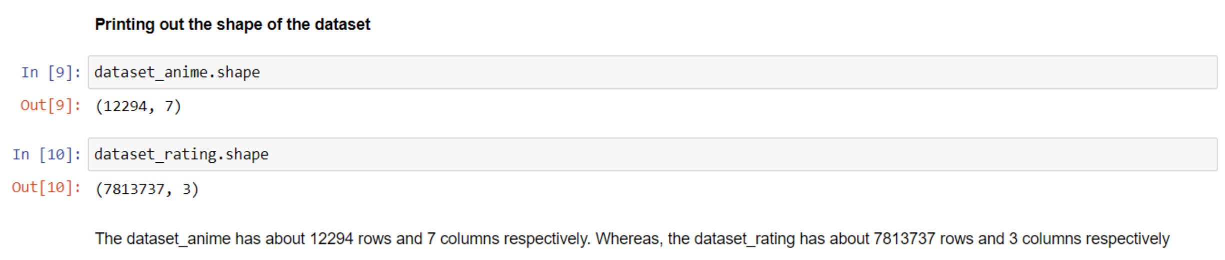
Members – number of community members involved in anime's group

- Features in rating.csv

user_id – represents the id of each user that rates an anime

anime_id – id for identifying the name of the anime in anime.csv file

Rating – rating of anime ranging from -1 to 10.

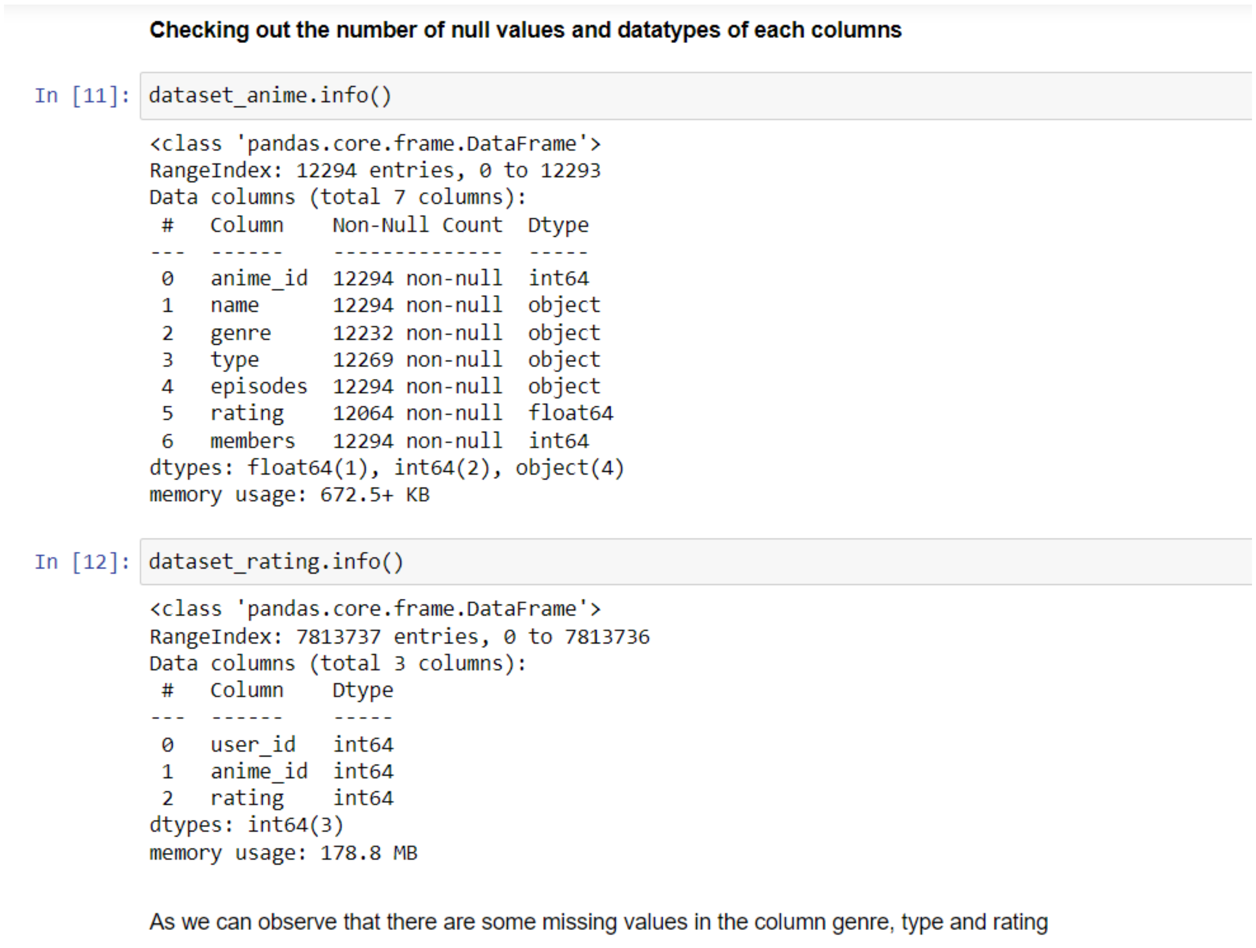


Printing out the shape of the dataset

```
In [9]: dataset_anime.shape
Out[9]: (12294, 7)

In [10]: dataset_rating.shape
Out[10]: (7813737, 3)
```

The dataset_anime has about 12294 rows and 7 columns respectively. Whereas, the dataset_rating has about 7813737 rows and 3 columns respectively

Figure 2: Checking out the shape of the datasets

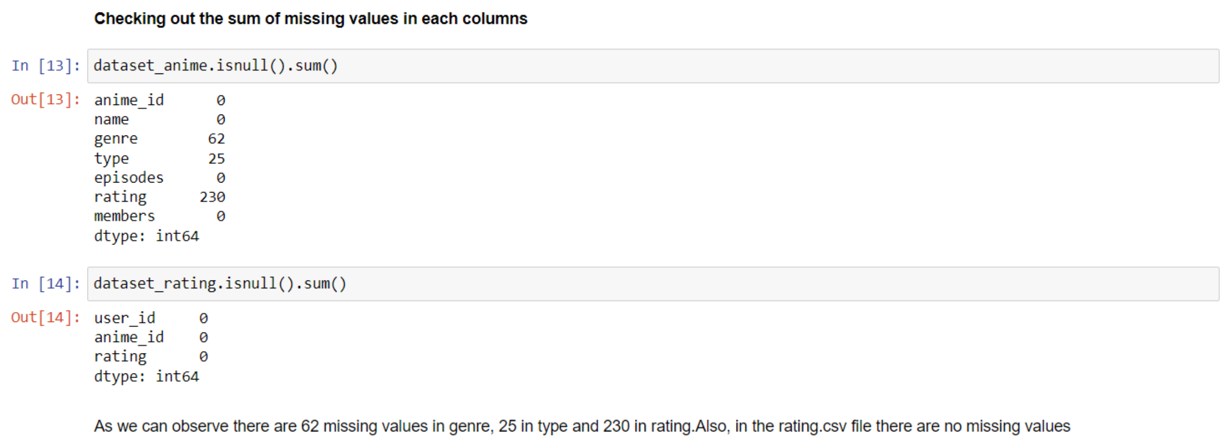


Checking out the number of null values and datatypes of each columns

```
In [11]: dataset_anime.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   anime_id  12294 non-null  int64  
 1   name      12294 non-null  object 
 2   genre     12232 non-null  object 
 3   type      12269 non-null  object 
 4   episodes  12294 non-null  object 
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64  
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB

In [12]: dataset_rating.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7813737 entries, 0 to 7813736
Data columns (total 3 columns):
 #   Column    Dtype
---  ------    -----
 0   user_id   int64
 1   anime_id  int64
 2   rating    int64
dtypes: int64(3)
memory usage: 178.8 MB
```

As we can observe that there are some missing values in the column genre, type and rating

Figure 3: Checking out the information of both the datasets


Checking out the sum of missing values in each columns

```
In [13]: dataset_anime.isnull().sum()

Out[13]: anime_id      0
         name          0
         genre        62
         type         25
         episodes      0
         rating      230
         members       0
         dtype: int64

In [14]: dataset_rating.isnull().sum()

Out[14]: user_id     0
         anime_id    0
         rating      0
         dtype: int64
```

As we can observe there are 62 missing values in genre, 25 in type and 230 in rating.Also, in the rating.csv file there are no missing values

Figure 4: Checking out the missing values sum in both the dataset

Implementation of MongoDB and python

Created a database named: Bigdata Coursework

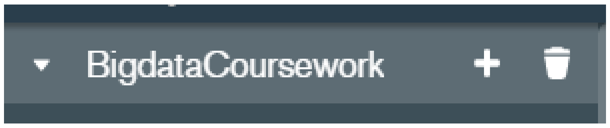
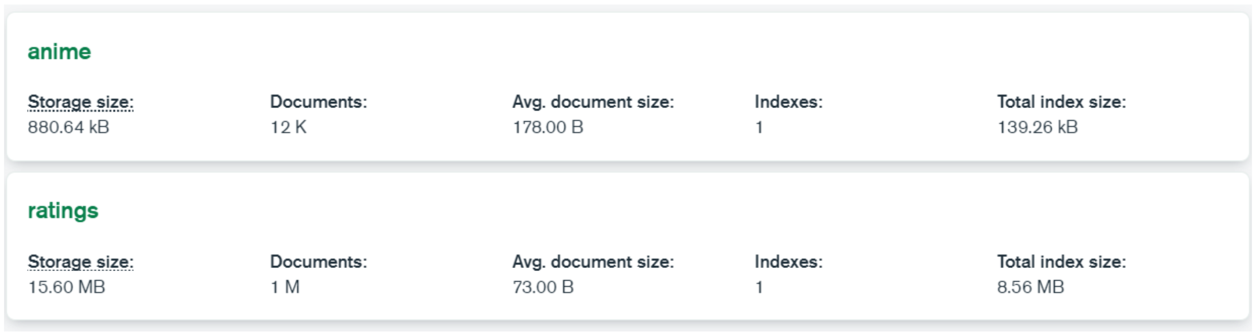


Figure 5: Screenshot created database in Mongodb

Created two collection named anime and ratings inside the database Bigdata Coursework


anime

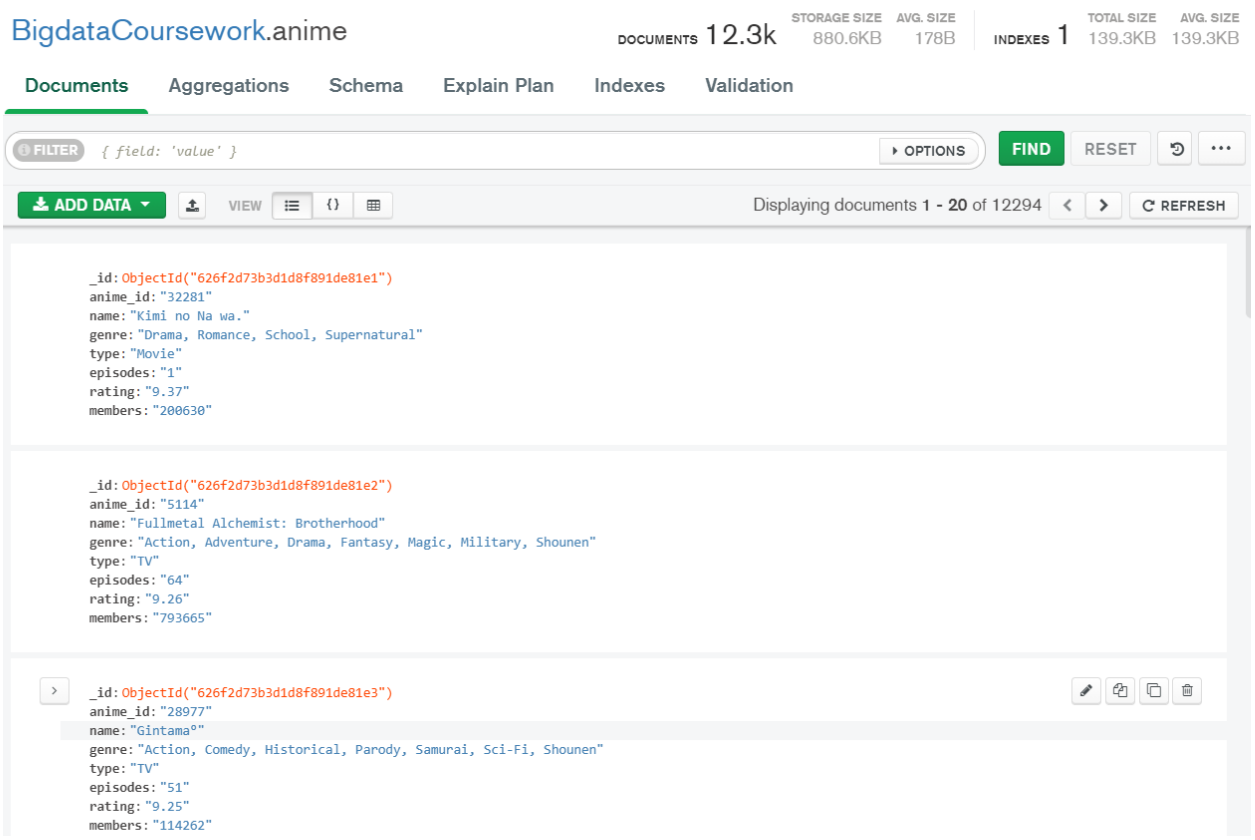
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
880.64 kB	12 K	178.00 B	1	139.26 kB

ratings

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
15.60 MB	1 M	73.00 B	1	8.56 MB

Figure 6: Screenshot of created collection in the Mongodb

Imported the csv file named anime.csv in the collection anime


BigdataCoursework.anime

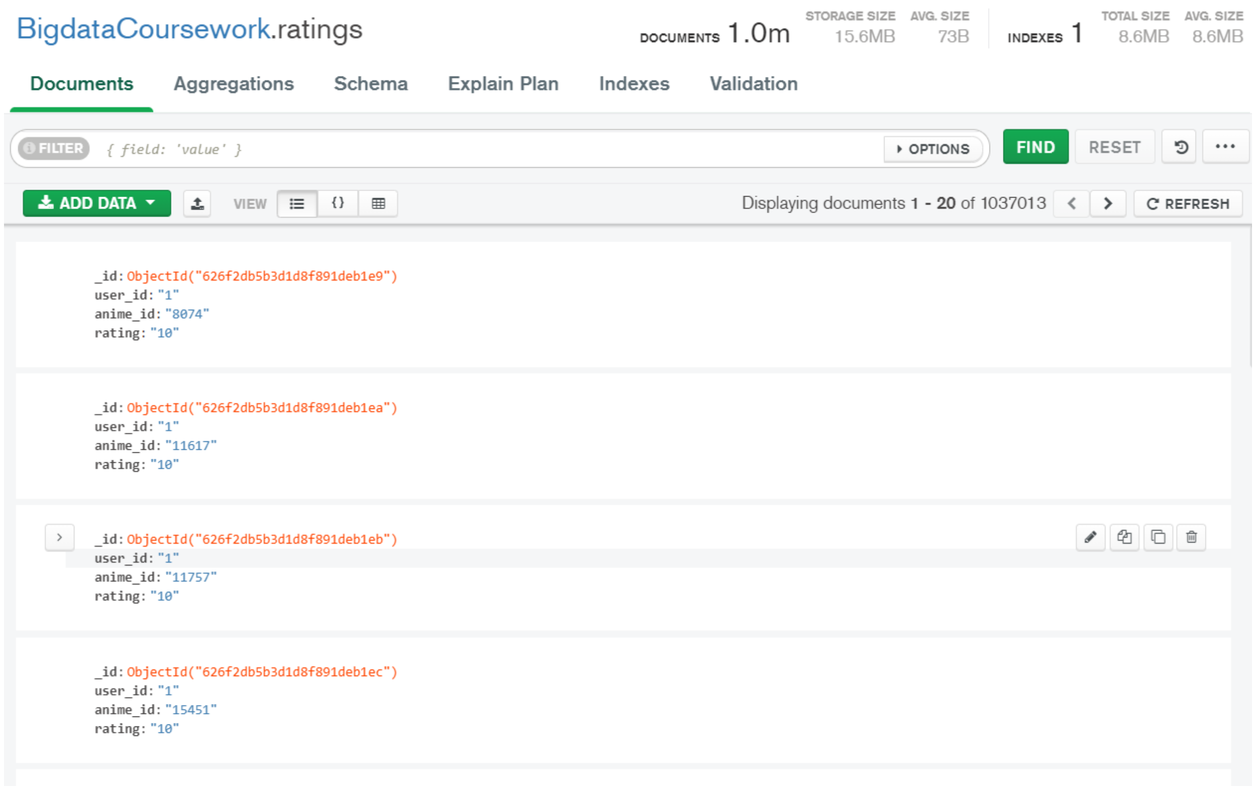
DOCUMENTS 12.3k STORAGE SIZE 880.6KB AVG. SIZE 178B INDEXES 1 TOTAL SIZE 139.3KB AVG. SIZE 139.3KB

Displaying documents 1 - 20 of 12294

```
_id: ObjectId("626f2d73b3d1d8f891de81e1")
anime_id: "32281"
name: "Kimi no Na wa."
genre: "Drama, Romance, School, Supernatural"
type: "Movie"
episodes: "1"
rating: "9.37"
members: "200630"

_id: ObjectId("626f2d73b3d1d8f891de81e2")
anime_id: "5114"
name: "Fullmetal Alchemist: Brotherhood"
genre: "Action, Adventure, Drama, Fantasy, Magic, Military, Shounen"
type: "TV"
episodes: "64"
rating: "9.26"
members: "793665"

_id: ObjectId("626f2d73b3d1d8f891de81e3")
anime_id: "28977"
name: "Gintama°"
genre: "Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen"
type: "TV"
episodes: "51"
rating: "9.25"
members: "114262"
```

Figure 7: Screenshot of the data import from anime.csv file

Imported the ratings.csv file inside the collection ratings

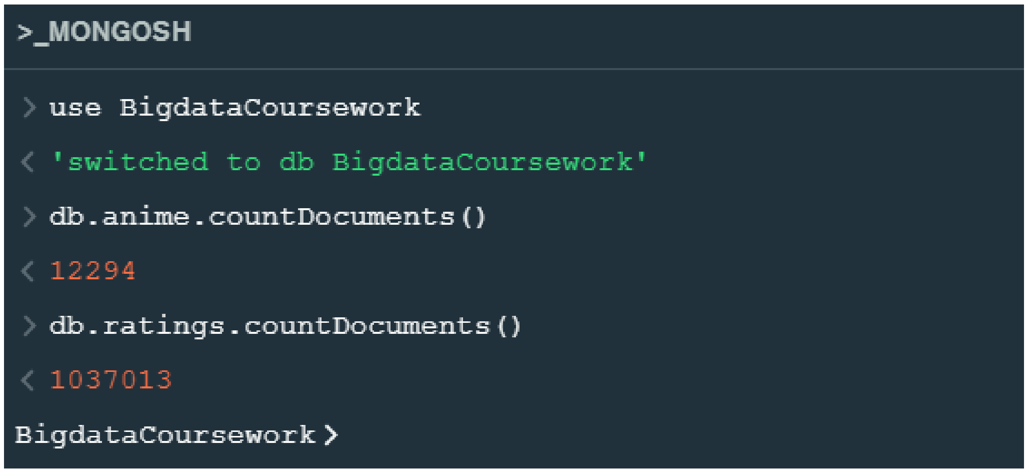


```
{ field: 'value' }
Displaying documents 1 - 20 of 1037013

_id: ObjectId("626f2db5b3d1d8f891deb1e9")
user_id: "1"
anime_id: "8074"
rating: "10"

_id: ObjectId("626f2db5b3d1d8f891deb1ea")
user_id: "1"
anime_id: "11617"
rating: "10"

_id: ObjectId("626f2db5b3d1d8f891deb1eb")
user_id: "1"
anime_id: "11757"
rating: "10"

_id: ObjectId("626f2db5b3d1d8f891deb1ec")
user_id: "1"
anime_id: "15451"
rating: "10"
```

Figure 8: Screenshot of the data import from rating.csv file

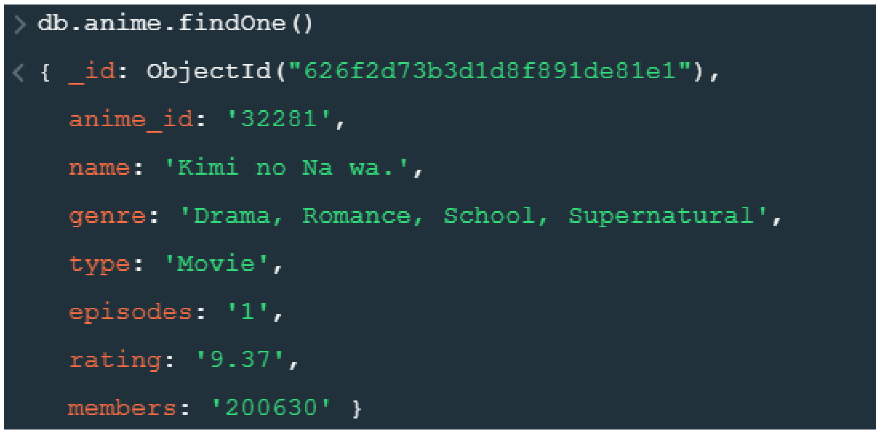
7.2. Data Analysis

Counting the number of documents present in both the anime and ratings collection.

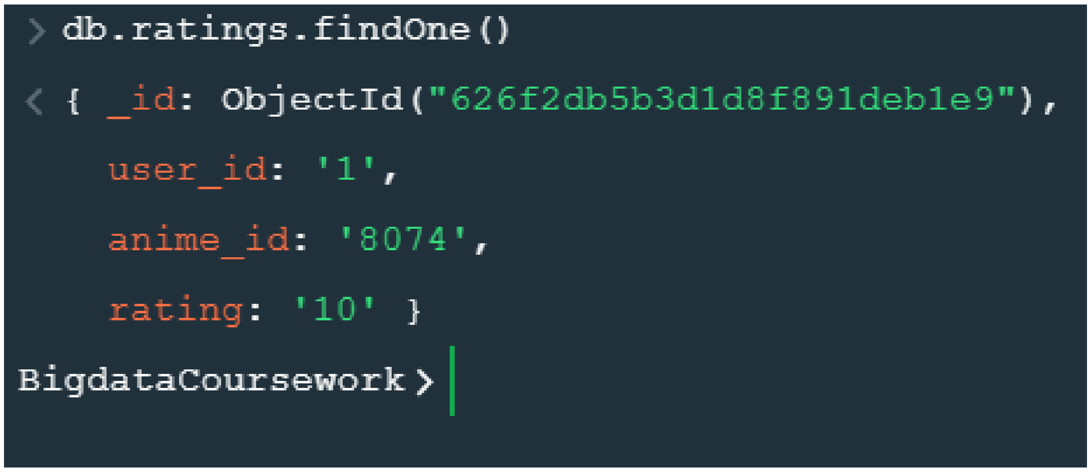


```
> use BigdataCoursework
< 'switched to db BigdataCoursework'
> db.anime.countDocuments()
< 12294
> db.ratings.countDocuments()
< 1037013
BigdataCoursework>
```

Showing one documents of each collection

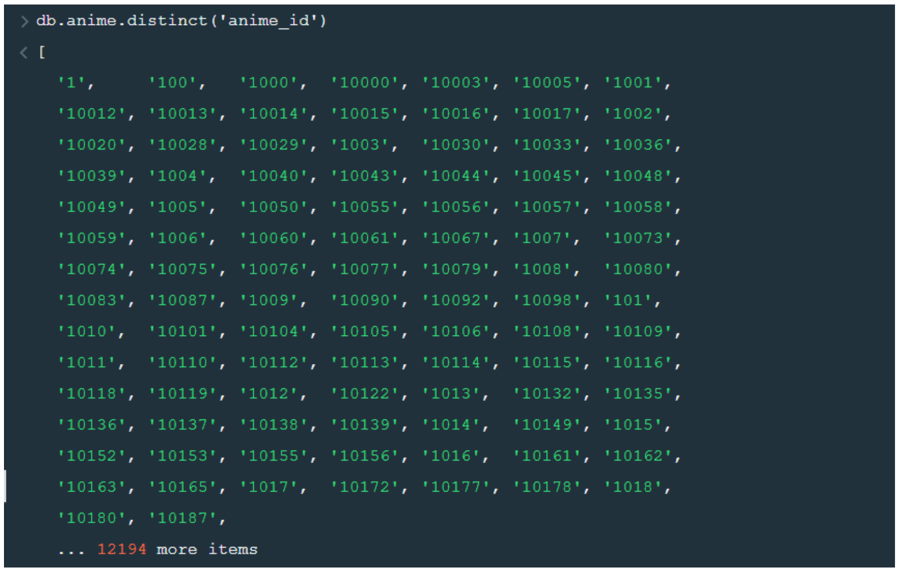


```
> db.anime.findOne()
< { _id: ObjectId("626f2d73b3d1d8f891de81e1"),
    anime_id: '32281',
    name: 'Kimi no Na wa.',
    genre: 'Drama, Romance, School, Supernatural',
    type: 'Movie',
    episodes: '1',
    rating: '9.37',
    members: '200630' }
```



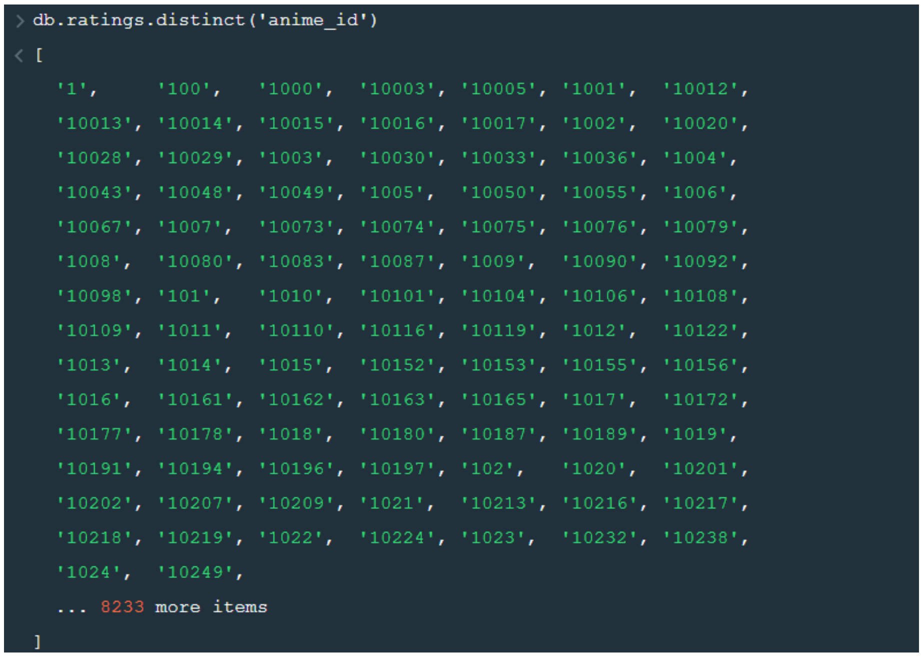
```
> db.ratings.findOne()
< { _id: ObjectId("626f2db5b3d1d8f891deb1e9"),
    user_id: '1',
    anime_id: '8074',
    rating: '10' }
BigdataCoursework >
```

Showing the unique values of collection anime in column anime_id



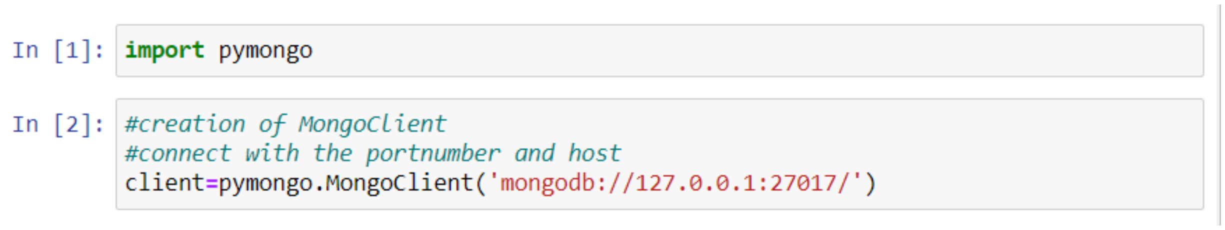
```
> db.anime.distinct('anime_id')
< [
    '1',     '100',   '1000',  '10000', '10003', '10005', '1001',
    '10012', '10013', '10014', '10015', '10016', '10017', '1002',
    '10020', '10028', '10029', '1003',  '10030', '10033', '10036',
    '10039', '1004',  '10040', '10043', '10044', '10045', '10048',
    '10049', '1005',  '10050', '10055', '10056', '10057', '10058',
    '10059', '1006',  '10060', '10061', '10067', '1007',  '10073',
    '10074', '10075', '10076', '10077', '10079', '1008',  '10080',
    '10083', '10087', '1009',  '10090', '10092', '10098', '101',
    '1010',  '10101', '10104', '10105', '10106', '10108', '10109',
    '1011',  '10110', '10112', '10113', '10114', '10115', '10116',
    '10118', '10119', '1012',  '10122', '1013',  '10132', '10135',
    '10136', '10137', '10138', '10139', '1014',  '10149', '1015',
    '10152', '10153', '10155', '10156', '1016',  '10161', '10162',
    '10163', '10165', '1017',  '10172', '10177', '10178', '1018',
    '10180', '10187',
    ... 12194 more items
```

Showing the unique values of collection rating in column anime_id



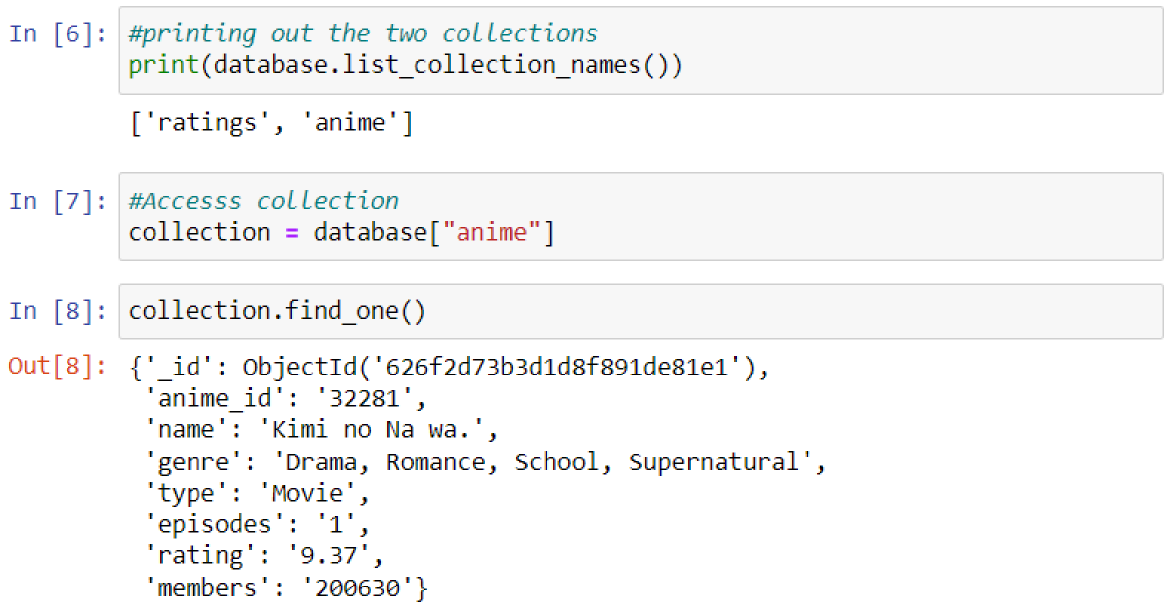
```
> db.ratings.distinct('anime_id')
< [
    '1',     '100',   '1000',  '10003', '10005', '1001',  '10012',
    '10013', '10014', '10015', '10016', '10017', '1002',  '10020',
    '10028', '10029', '1003',  '10030', '10033', '10036', '1004',
    '10043', '10048', '10049', '1005',  '10050', '10055', '1006',
    '10067', '1007',  '10073', '10074', '10075', '10076', '10079',
    '1008',  '10080', '10083', '10087', '1009',  '10090', '10092',
    '10098', '101',   '1010',  '10101', '10104', '10106', '10108',
    '10109', '1011',  '10110', '10116', '10119', '1012',  '10122',
    '1013',  '1014',  '1015',  '10152', '10153', '10155', '10156',
    '1016',  '10161', '10162', '10163', '10165', '1017',  '10172',
    '10177', '10178', '1018',  '10180', '10187', '10189', '1019',
    '10191', '10194', '10196', '10197', '102',   '1020',  '10201',
    '10202', '10207', '10209', '1021',  '10213', '10216', '10217',
    '10218', '10219', '1022',  '10224', '1023',  '10232', '10238',
    '1024',  '10249',
    ... 8233 more items
  ]
```

Connecting mongodb with jupyter notebook

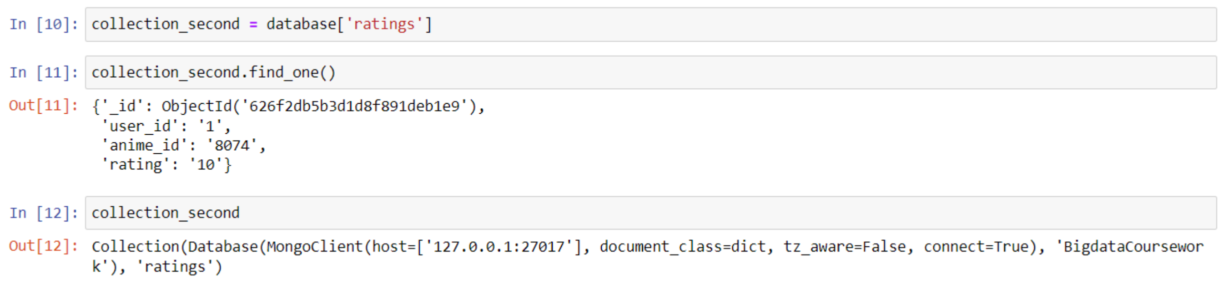


```
In [1]: import pymongo

In [2]: #creation of MongoClient
        #connect with the portnumber and host
        client=pymongo.MongoClient('mongodb://127.0.0.1:27017/')
```

Accessing the database 'BigdataCoursework' and the collection present inside them

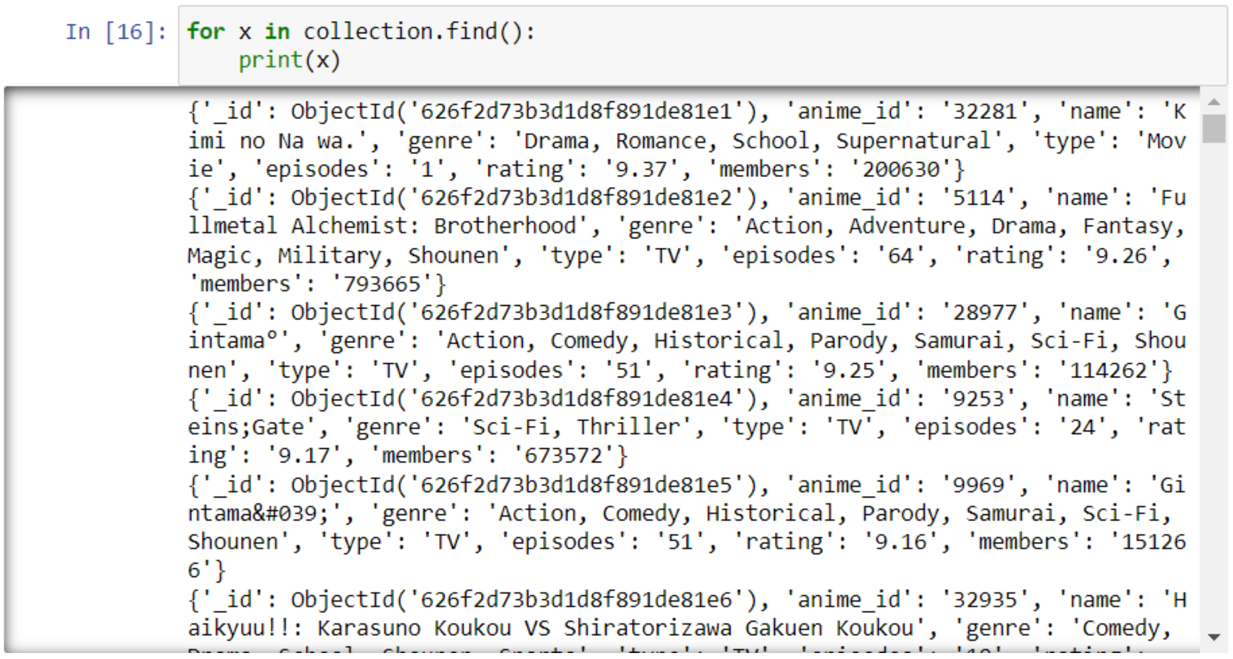


```
In [6]: #printing out the two collections
        print(database.list_collection_names())

        ['ratings', 'anime']

In [7]: #Accesss collection
        collection = database["anime"]

In [8]: collection.find_one()

Out[8]: {'_id': ObjectId('626f2d73b3d1d8f891de81e1'),
         'anime_id': '32281',
         'name': 'Kimi no Na wa.',
         'genre': 'Drama, Romance, School, Supernatural',
         'type': 'Movie',
         'episodes': '1',
         'rating': '9.37',
         'members': '200630'}
```



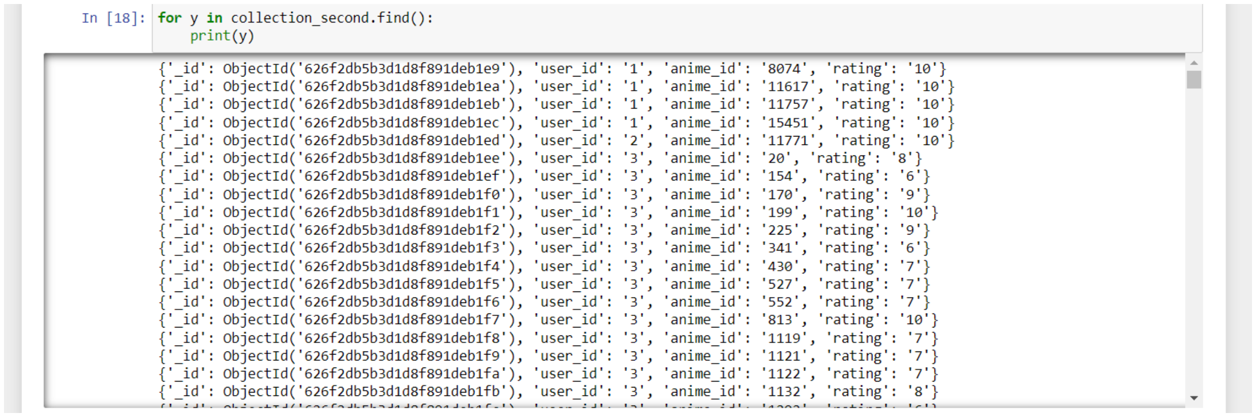
```
In [10]: collection_second = database['ratings']

In [11]: collection_second.find_one()

Out[11]: {'_id': ObjectId('626f2db5b3d1d8f891deb1e9'),
          'user_id': '1',
          'anime_id': '8074',
          'rating': '10'}

In [12]: collection_second

Out[12]: Collection(Database(MongoClient(host=['127.0.0.1:27017'], document_class=dict, tz_aware=False, connect=True), 'BigdataCoursewor
         k'), 'ratings')
```

Printing out all the documents of collection anime



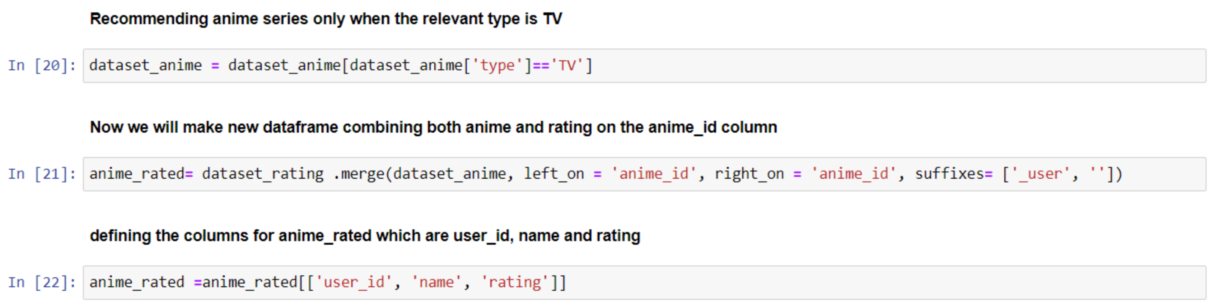
```
In [16]: for x in collection.find():
             print(x)

{'_id': ObjectId('626f2d73b3d1d8f891de81e1'), 'anime_id': '32281', 'name': 'K
imi no Na wa.', 'genre': 'Drama, Romance, School, Supernatural', 'type': 'Mov
ie', 'episodes': '1', 'rating': '9.37', 'members': '200630'}
{'_id': ObjectId('626f2d73b3d1d8f891de81e2'), 'anime_id': '5114', 'name': 'Fu
llmetal Alchemist: Brotherhood', 'genre': 'Action, Adventure, Drama, Fantasy,
Magic, Military, Shounen', 'type': 'TV', 'episodes': '64', 'rating': '9.26',
'members': '793665'}
{'_id': ObjectId('626f2d73b3d1d8f891de81e3'), 'anime_id': '28977', 'name': 'G
intama°', 'genre': 'Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shou
nen', 'type': 'TV', 'episodes': '51', 'rating': '9.25', 'members': '114262'}
{'_id': ObjectId('626f2d73b3d1d8f891de81e4'), 'anime_id': '9253', 'name': 'St
eins;Gate', 'genre': 'Sci-Fi, Thriller', 'type': 'TV', 'episodes': '24', 'rat
ing': '9.17', 'members': '673572'}
{'_id': ObjectId('626f2d73b3d1d8f891de81e5'), 'anime_id': '9969', 'name': 'Gi
ntama&#039;', 'genre': 'Action, Comedy, Historical, Parody, Samurai, Sci-Fi,
Shounen', 'type': 'TV', 'episodes': '51', 'rating': '9.16', 'members': '15126
6'}
{'_id': ObjectId('626f2d73b3d1d8f891de81e6'), 'anime_id': '32935', 'name': 'H
aikyuu!!: Karasuno Koukou VS Shiratorizawa Gakuen Koukou', 'genre': 'Comedy,
```

Printing out all the documents of collection ratings



```
In [18]: for y in collection_second.find():
             print(y)
```

```
{'_id': ObjectId('626f2db5b3d1d8f891deb1e9'), 'user_id': '1', 'anime_id': '8074', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1ea'), 'user_id': '1', 'anime_id': '11617', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1eb'), 'user_id': '1', 'anime_id': '11757', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1ec'), 'user_id': '1', 'anime_id': '15451', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1ed'), 'user_id': '2', 'anime_id': '11771', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1ee'), 'user_id': '3', 'anime_id': '20', 'rating': '8'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1ef'), 'user_id': '3', 'anime_id': '154', 'rating': '6'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f0'), 'user_id': '3', 'anime_id': '170', 'rating': '9'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f1'), 'user_id': '3', 'anime_id': '199', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f2'), 'user_id': '3', 'anime_id': '225', 'rating': '9'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f3'), 'user_id': '3', 'anime_id': '341', 'rating': '6'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f4'), 'user_id': '3', 'anime_id': '430', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f5'), 'user_id': '3', 'anime_id': '527', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f6'), 'user_id': '3', 'anime_id': '552', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f7'), 'user_id': '3', 'anime_id': '813', 'rating': '10'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f8'), 'user_id': '3', 'anime_id': '1119', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1f9'), 'user_id': '3', 'anime_id': '1121', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1fa'), 'user_id': '3', 'anime_id': '1122', 'rating': '7'}
{'_id': ObjectId('626f2db5b3d1d8f891deb1fb'), 'user_id': '3', 'anime_id': '1132', 'rating': '8'}
```



Recommending anime series only when the relevant type is TV

```
In [20]: dataset_anime = dataset_anime[dataset_anime['type']=='TV']
```

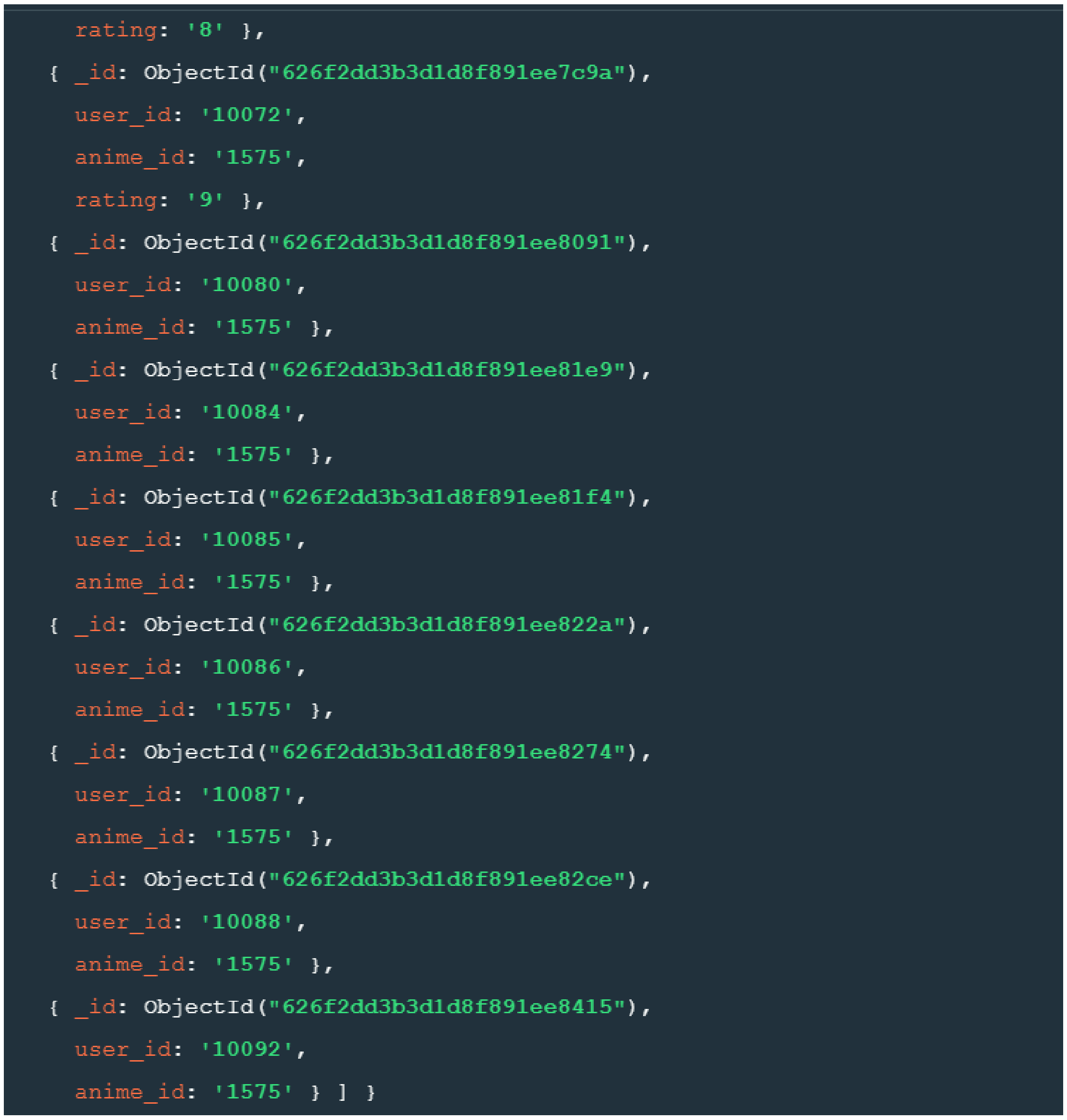
Now we will make new dataframe combining both anime and rating on the anime_id column

```
In [21]: anime_rated= dataset_rating .merge(dataset_anime, left_on = 'anime_id', right_on = 'anime_id', suffixes= ['_user', ''])
```

defining the columns for anime_rated which are user_id, name and rating

```
In [22]: anime_rated =anime_rated[['user_id', 'name', 'rating']]
```

Figure 9: Joining anime and rating dataset as their anime_id is similar



```
    rating: '8' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee7c9a"),
    user_id: '10072',
    anime_id: '1575',
    rating: '9' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee8091"),
    user_id: '10080',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee81e9"),
    user_id: '10084',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee81f4"),
    user_id: '10085',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee822a"),
    user_id: '10086',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee8274"),
    user_id: '10087',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee82ce"),
    user_id: '10088',
    anime_id: '1575' },
  { _id: ObjectId("626f2dd3b3d1d8f891ee8415"),
    user_id: '10092',
    anime_id: '1575' } ] }
```

Figure 10: Performing join using lookup in anime and rating

7.3. Data Visualization

```
In [74]: import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns
         # plot distribution in matplotlib
         ratings_sorted = sorted(list(zip(ratings_y.index, ratings_y)))
         plt.bar([r[0] for r in ratings_sorted], [r[1] for r in ratings_sorted], color='cyan')
         plt.xlabel("Rating")
         plt.ylabel("# of Ratings")
         plt.title("Distribution of Ratings")
         plt.show()
```

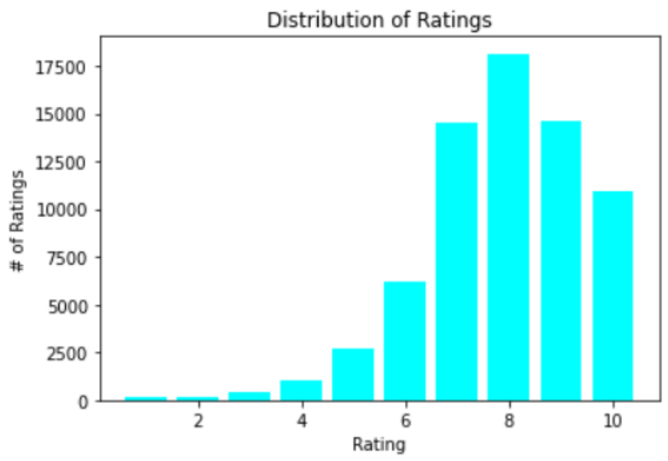


Figure 11: Figure for distribution of ratings

```
In [75]: ratings_per_user = sample['user_id'].value_counts()
         ratings_per_user = sorted(list(zip(ratings_per_user.index, ratings_per_user)))
         plt.bar([r[0] for r in ratings_per_user], [r[1] for r in ratings_per_user], color='pink')
         plt.xlabel('User IDs')
         plt.ylabel('# of Reviews')
         plt.title('Number of Reviews per User')
         plt.show()
```

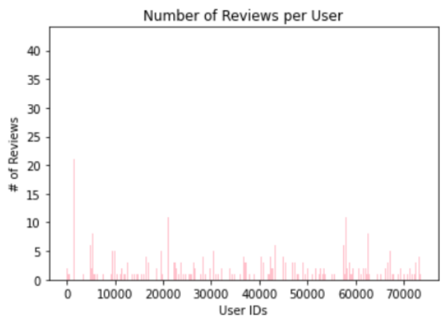
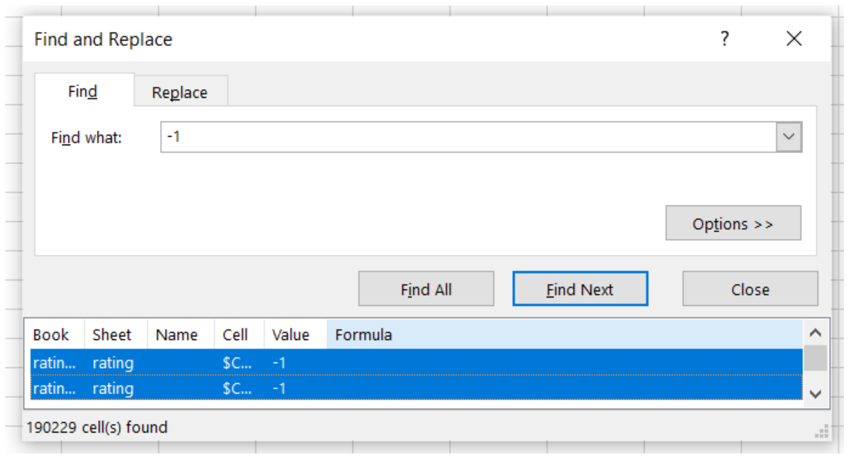


Figure 12: Figure for number of reviews per user

7.4. Cleaning the data

a. Removing all the -1 values from the rating.csv file in the excel

Since rating should be from 1 to 10, -1 could be no ratings which does not seems like a useful value in the rating csv.

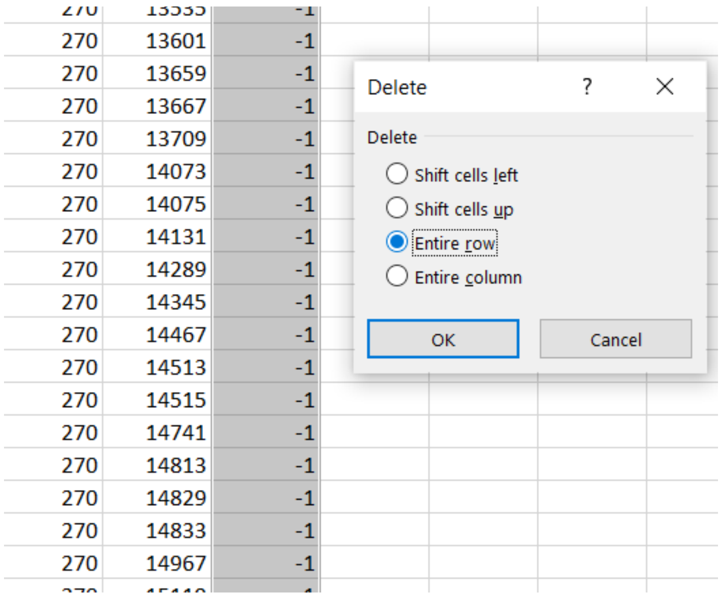


Find what: -1

Book	Sheet	Name	Cell	Value	Formula
ratin...	rating		$C...	-1	
ratin...	rating		$C...	-1	

190229 cell(s) found

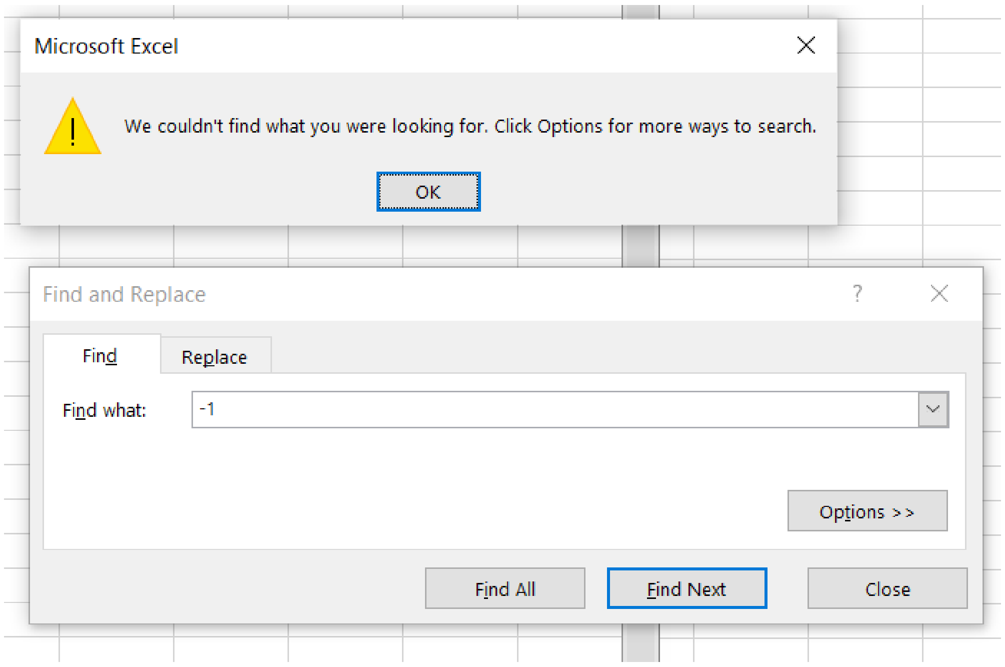
Figure 13: Screenshot of finding all the -1 values in the excel file



270	13601	-1
270	13659	-1
270	13667	-1
270	13709	-1
270	14073	-1
270	14075	-1
270	14131	-1
270	14289	-1
270	14345	-1
270	14467	-1
270	14513	-1
270	14515	-1
270	14741	-1
270	14813	-1
270	14829	-1
270	14833	-1
270	14967	-1

Delete: Shift cells left / Shift cells up / [✓ Entire row] / Entire column

Figure 14: Deleting the rows with the value -1


Microsoft Excel

We couldn't find what you were looking for. Click Options for more ways to search.

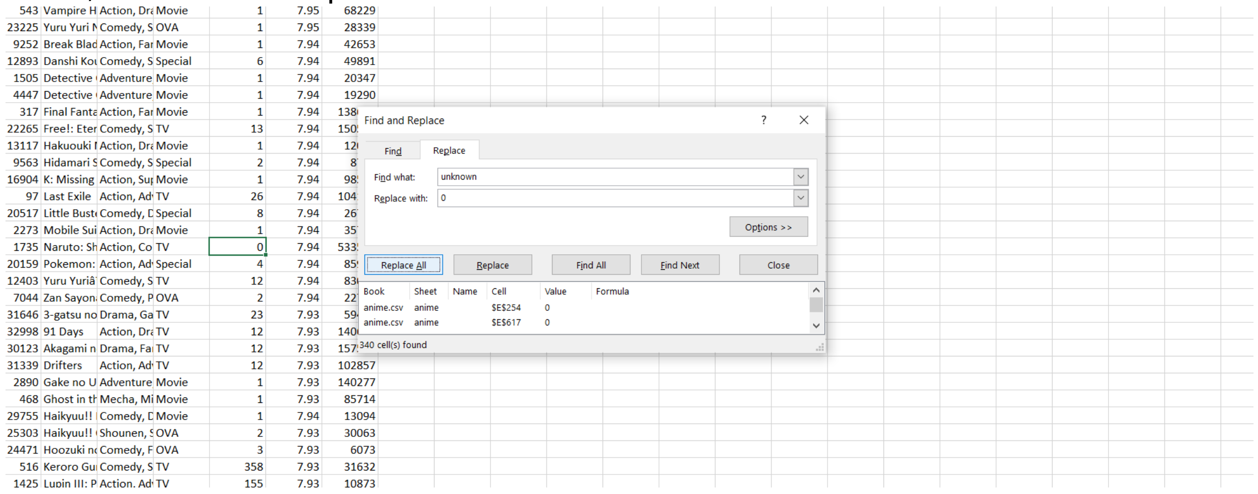
Find and Replace

Find what: -1

Figure 15: Checking if the -1 value exists or not in the ratings file in excel

b. Replacing all the unknown values in the episode column of anime.csv file in excel

Since, the number of episode can be a useful value.


543	Vampire H	Action, Dra	Movie	1	7.95	68229
23225	Yuru Yuri N	Comedy, S	OVA	1	7.95	28339
9252	Break Blad	Action, Far	Movie	1	7.94	42653
12893	Danshi Kou	Comedy, S	Special	6	7.94	49891
1505	Detective (	Adventure	Movie	1	7.94	20347
4447	Detective (	Adventure	Movie	1	7.94	19290
317	Final Fanta	Action, Far	Movie	1	7.94	138
22265	Free!: Eter	Comedy, S	TV	13	7.94	150
13117	Hakuouki I	Action, Dra	Movie	1	7.94	12
9563	Hidamari S	Comedy, S	Special	2	7.94	8
16904	K: Missing	Action, Sup	Movie	1	7.94	98
97	Last Exile	Action, Adv	TV	26	7.94	104
20517	Little Buste	Comedy, D	Special	8	7.94	26
2273	Mobile Sui	Action, Dra	Movie	1	7.94	35
1735	Naruto: Sh	Action, Co	TV	0	7.94	533
20159	Pokemon:	Action, Adv	Special	4	7.94	85
12403	Yuru Yuriâ	Comedy, S	TV	12	7.94	83
7044	Zan Sayon	Comedy, P	OVA	2	7.94	22
31646	3-gatsu no	Drama, Ga	TV	23	7.93	59
32998	91 Days	Action, Dra	TV	12	7.93	140
30123	Akagami n	Drama, Far	TV	12	7.93	157
31339	Drifters	Action, Adv	TV	12	7.93	102857
2890	Gake no U	Adventure	Movie	1	7.93	140277
468	Ghost in th	Mecha, Mi	Movie	1	7.93	85714
29755	Haikyuu!! I	Comedy, D	Movie	1	7.94	13094
25303	Haikyuu!! (	Shounen, S	OVA	2	7.93	30063
24471	Hoozuki no	Comedy, F	OVA	3	7.93	6073
516	Keroro Gu	Comedy, S	TV	358	7.93	31632
1425	Lupin III: P	Action, Adv	TV	155	7.93	10873

Find and Replace

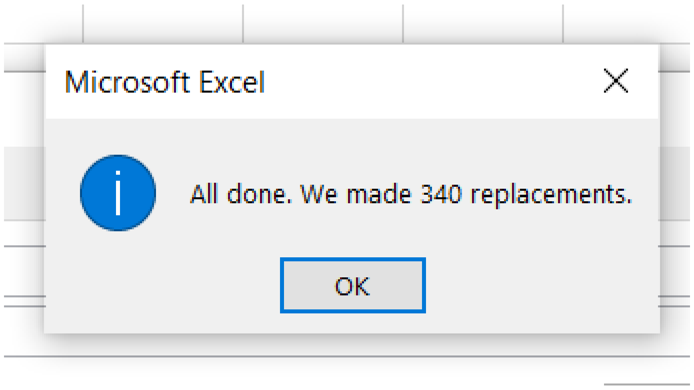
Find what: unknown

Replace with: 0

Book	Sheet	Name	Cell	Value	Formula
anime.csv	anime		E254	0	
anime.csv	anime		E617	0	

340 cell(s) found

Figure 16: Replacing the unknown row value episode to 0 in the column episode in Anime.csv file in excel

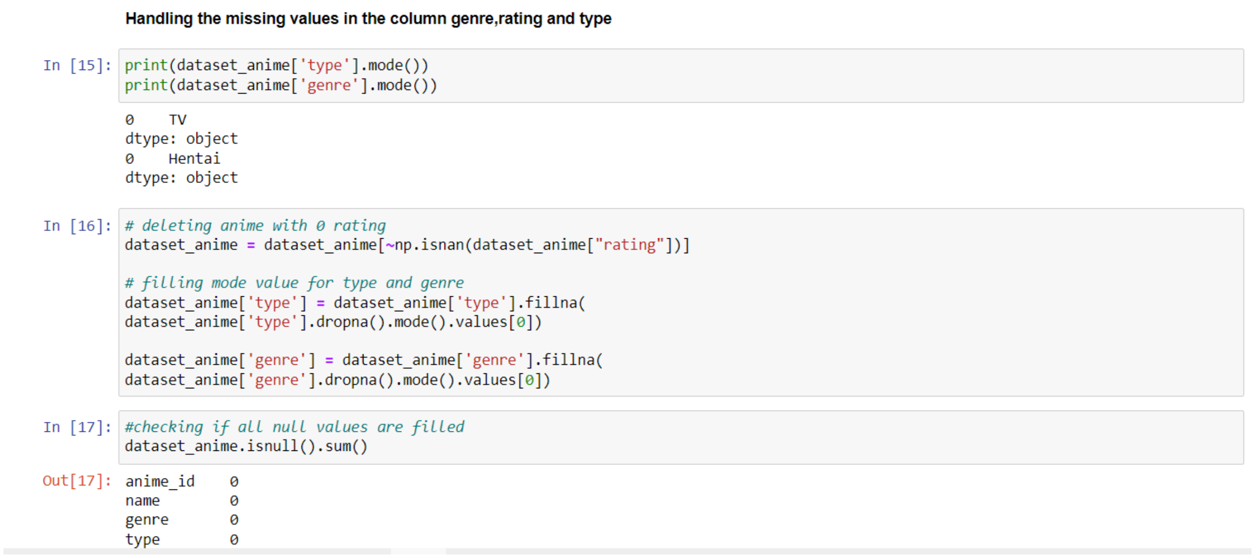


All done. We made 340 replacements.

Figure 17: Successfully replaced the unknown value with 0 in excel

Replaced all the unknown values of the episodes as 0.

C. Handling the missing values



Handling the missing values in the column genre,rating and type

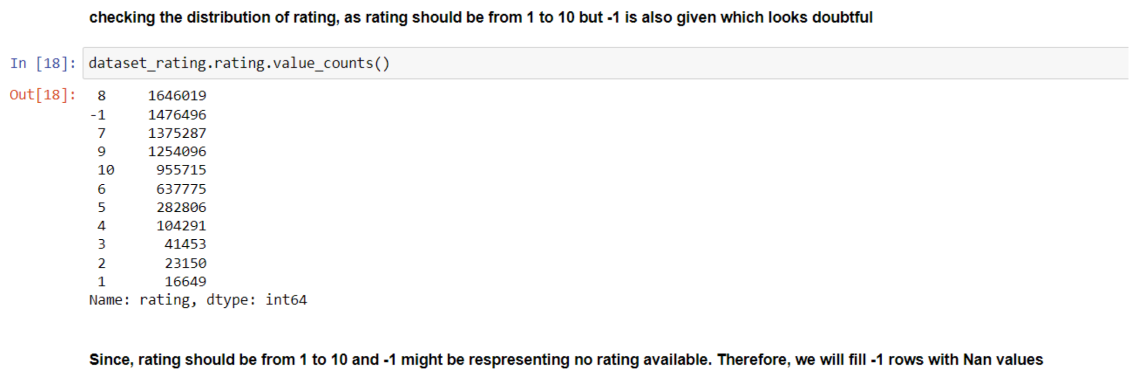
```
In [15]: print(dataset_anime['type'].mode())
         print(dataset_anime['genre'].mode())

0    TV
dtype: object
0    Hentai
dtype: object
```

```
In [16]: # deleting anime with 0 rating
         dataset_anime = dataset_anime[~np.isnan(dataset_anime["rating"])]

         # filling mode value for type and genre
         dataset_anime['type'] = dataset_anime['type'].fillna(
         dataset_anime['type'].dropna().mode().values[0])

         dataset_anime['genre'] = dataset_anime['genre'].fillna(
         dataset_anime['genre'].dropna().mode().values[0])
```

```
In [17]: #checking if all null values are filled
         dataset_anime.isnull().sum()

Out[17]: anime_id    0
         name        0
         genre       0
         type        0
```

Figure 18: Handling the missing values of both datasets in python

d. Replacing -1 with Nan values

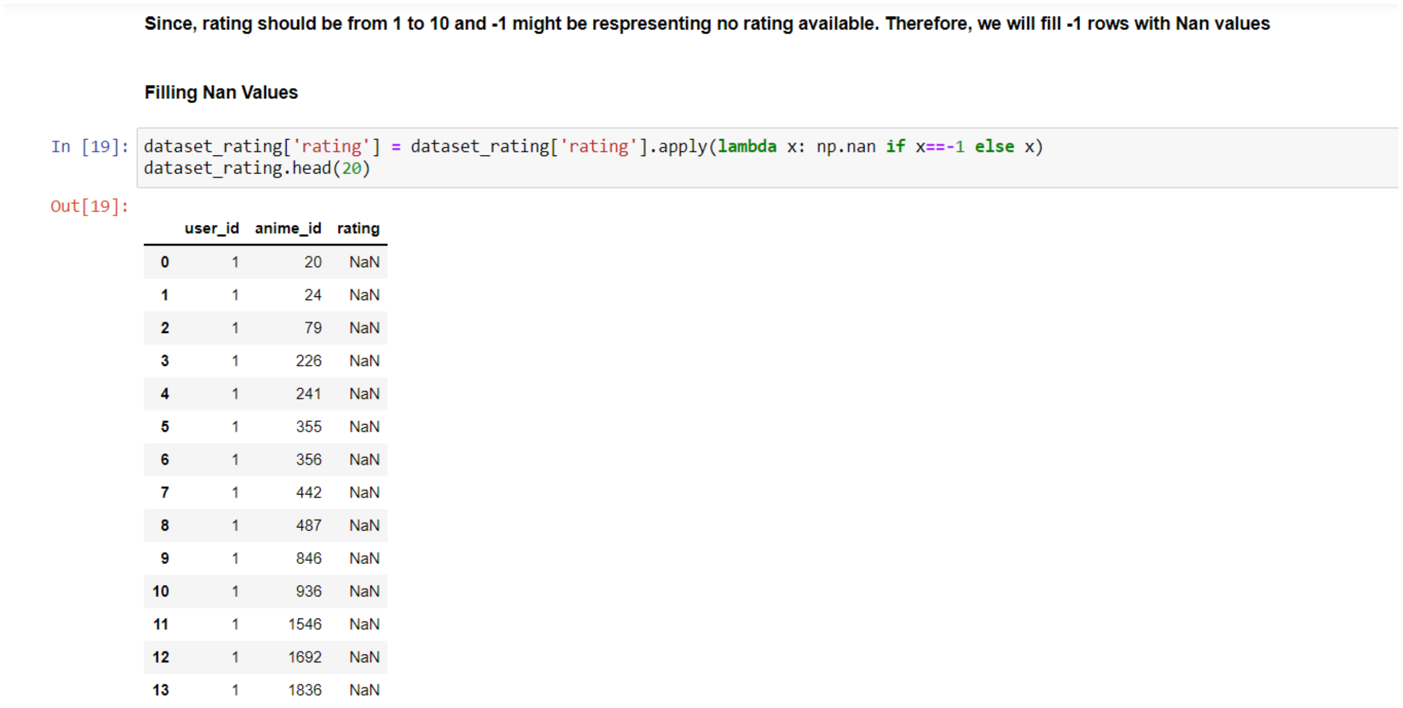


checking the distribution of rating, as rating should be from 1 to 10 but -1 is also given which looks doubtful

```
In [18]: dataset_rating.rating.value_counts()
```

```
Out[18]:  8     1646019
         -1     1476496
          7     1375287
          9     1254096
          10     955715
          6      637775
          5      282806
          4      104291
          3       41453
          2       23150
          1       16649
         Name: rating, dtype: int64
```

Since, rating should be from 1 to 10 and -1 might be respresenting no rating available. Therefore, we will fill -1 rows with Nan values


Since, rating should be from 1 to 10 and -1 might be respresenting no rating available. Therefore, we will fill -1 rows with Nan values

Filling Nan Values

```
In [19]: dataset_rating['rating'] = dataset_rating['rating'].apply(lambda x: np.nan if x==-1 else x)
         dataset_rating.head(20)
```

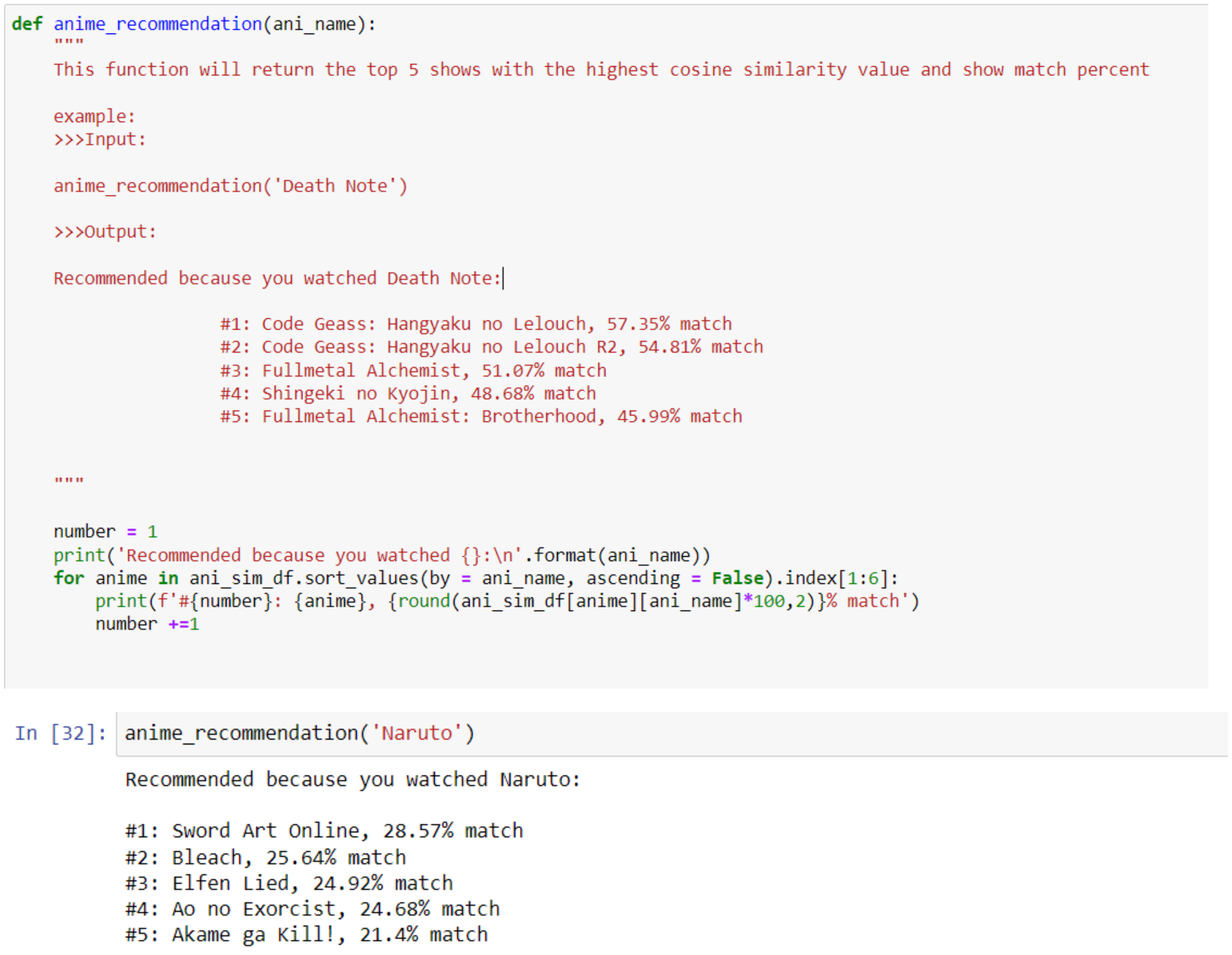
Out[19]:

	user_id	anime_id	rating
0	1	20	NaN
1	1	24	NaN
2	1	79	NaN
3	1	226	NaN
4	1	241	NaN
5	1	355	NaN
6	1	356	NaN
7	1	442	NaN
8	1	487	NaN
9	1	846	NaN
10	1	936	NaN
11	1	1546	NaN
12	1	1692	NaN
13	1	1836	NaN

Figure 19: Filling Nan values in all the -1 values in rating.csv file

7.5. Choosing the best model

Cosine similarity

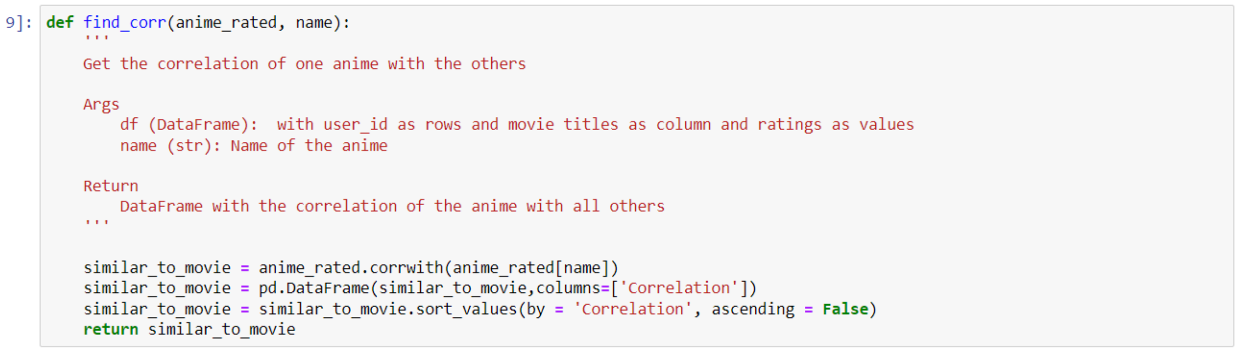


```
def anime_recommendation(ani_name):
    """
    This function will return the top 5 shows with the highest cosine similarity value and show match percent

    example:
    >>>Input:

    anime_recommendation('Death Note')

    >>>Output:

    Recommended because you watched Death Note:

                    #1: Code Geass: Hangyaku no Lelouch, 57.35% match
                    #2: Code Geass: Hangyaku no Lelouch R2, 54.81% match
                    #3: Fullmetal Alchemist, 51.07% match
                    #4: Shingeki no Kyojin, 48.68% match
                    #5: Fullmetal Alchemist: Brotherhood, 45.99% match


    """

    number = 1
    print('Recommended because you watched {}:\n'.format(ani_name))
    for anime in ani_sim_df.sort_values(by = ani_name, ascending = False).index[1:6]:
        print(f'#{number}: {anime}, {round(ani_sim_df[anime][ani_name]*100,2)}% match')
        number +=1
```

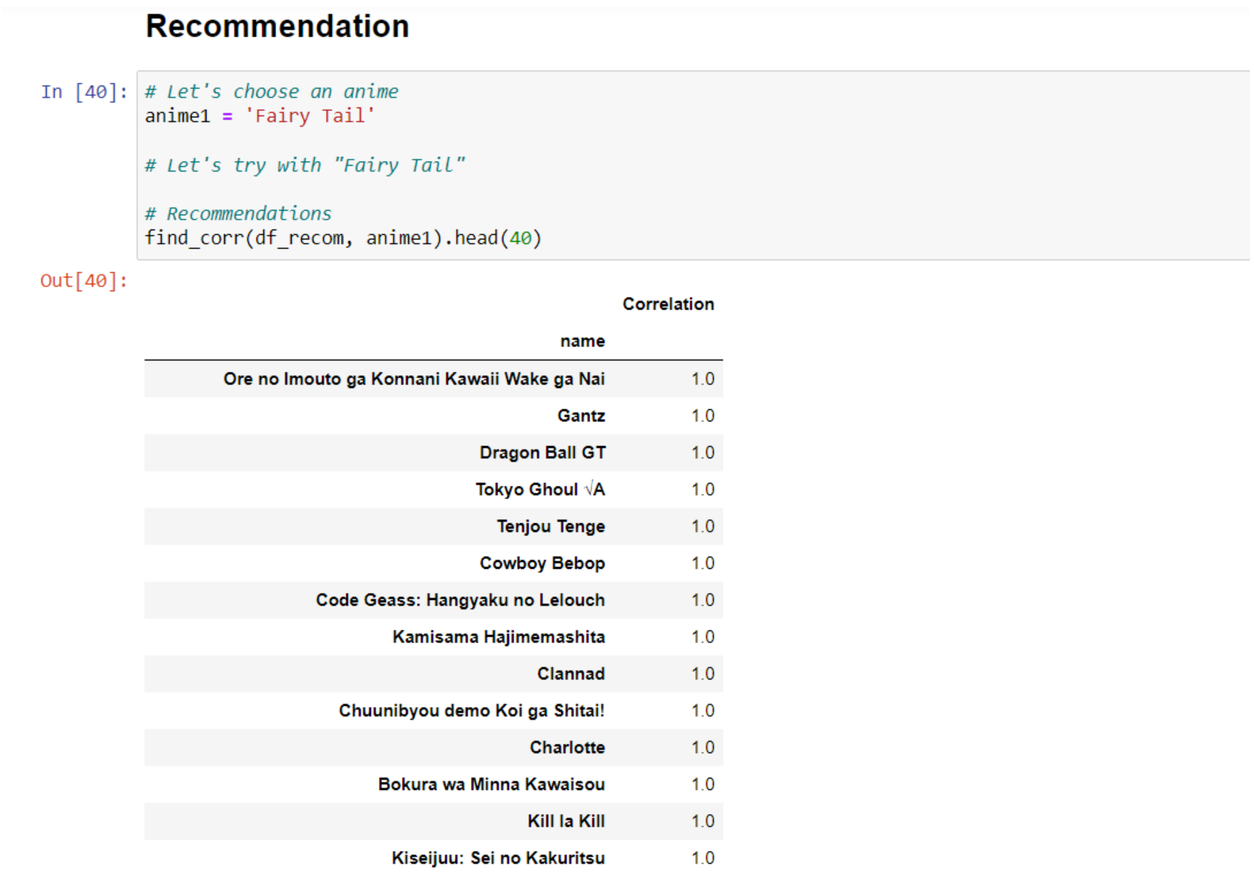
```
In [32]: anime_recommendation('Naruto')

         Recommended because you watched Naruto:

         #1: Sword Art Online, 28.57% match
         #2: Bleach, 25.64% match
         #3: Elfen Lied, 24.92% match
         #4: Ao no Exorcist, 24.68% match
         #5: Akame ga Kill!, 21.4% match
```

Figure 20: Cosine similarity function and recommended result screenshot

Collaborative Recommendation System



```
9]: def find_corr(anime_rated, name):
        '''
        Get the correlation of one anime with the others

        Args
            df (DataFrame):  with user_id as rows and movie titles as column and ratings as values
            name (str): Name of the anime

        Return
            DataFrame with the correlation of the anime with all others
        '''

        similar_to_movie = anime_rated.corrwith(anime_rated[name])
        similar_to_movie = pd.DataFrame(similar_to_movie,columns=['Correlation'])
        similar_to_movie = similar_to_movie.sort_values(by = 'Correlation', ascending = False)
        return similar_to_movie
```



Recommendation

```
In [40]: # Let's choose an anime
         anime1 = 'Fairy Tail'

         # Let's try with "Fairy Tail"

         # Recommendations
         find_corr(df_recom, anime1).head(40)
```

Out[40]:

name	Correlation
Ore no Imouto ga Konnani Kawaii Wake ga Nai	1.0
Gantz	1.0
Dragon Ball GT	1.0
Tokyo Ghoul √A	1.0
Tenjou Tenge	1.0
Cowboy Bebop	1.0
Code Geass: Hangyaku no Lelouch	1.0
Kamisama Hajimemashita	1.0
Clannad	1.0
Chuunibyou demo Koi ga Shitai!	1.0
Charlotte	1.0
Bokura wa Minna Kawaisou	1.0
Kill la Kill	1.0
Kiseijuu: Sei no Kakuritsu	1.0

Figure 21: Screenshot of function of collaborative filtering and its recommended result

As for the observation and result obtained, both the models are recommending properly.

8. Conclusion

Firstly, we can conclude that if the dataset is learned properly, we can gain good insights from it. We must be careful while pre-processing the data, as we might miss some meaningful implementation. If we are successful to pre-process the data, which includes removing the useless columns, changing the datatypes, removing outliers, and filling the null or missing value, we will end up getting an exact result as we have expected.

Secondly, big data plays a very essential role in managing large data which is impossible to handle using the traditional method. Search engine technology is becoming popular due to its ability to provide users with things they want to search for.

Lastly, in the recommendation engine, collaborative filtering is used, which recommends users' with the most popular rating anime, whereas, cosine similarity suggests similar anime based on nearest distance, and content-based filtering uses the genre of anime to recommend anime to the users'. Content-based filtering had some drawbacks that were covered by collaborative filtering. The need for a recommendation system is increasing as users' do find it difficult to search for the anime, or movies that match their taste, as there are lots of options to choose from. There are also some limitations when a recommendation system is built if a recommendation system is built using a year-old dataset, the system faces trouble with recommending consequently affecting the accuracy score. It is suggested that datasets should be updated according to the new work of anime, movies, and users' ratings.

9. Reference

Ahmed, I., 2020. *Astera.* [Online]
Available at: https://www.astera.com/type/blog/what-is-data-warehousing/
[Accessed 29 April 2022].

Drivas, I. C., Sakas, P. D., Giannakopoulos, G. A. & Manessi, D. K.-., 2020. Big Data Analytics for Search Engine Optimization. *MDPI*, 2 April, pp. 1-22.

Education, I. C., 2021. *IBM.* [Online]
Available at: https://www.ibm.com/cloud/learn/data-mining
[Accessed 29 April 2022].

Girsang, A. S., Faruq, B. A., Herlianto, H. R. & Simbolon, S., 2020. Collaborative Recommendation System in Users of Anime Flims. *Research Gate*, June, pp. 1-6.

Javed, M., 2020. *towardsdatascience.* [Online]
Available at: https://towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599
[Accessed 30 April 2022].

Knauff, J., 2018. *SEJ.* [Online]
Available at: https://www.searchenginejournal.com/search-engine-technology/256223/
[Accessed 29 April 2022].

Ota, S. et al., 2021. AniReco: Japanese Animne Recommendation System. *ResearchGate*, December, pp. 217-225.

Recommendation System, 2021. *Recommendation System.* [Online]
Available at: https://developers.google.com/machine-learning/recommendation/collaborative/basics
[Accessed 30 April 2022].

Siddharth_s, 2020. *Analytics Vidhya.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2020/11/what-is-big-data-a-quick-introduction-for-analytics-and-data-engineering-beginners/
[Accessed 29 April 2022].

Tableau, 2022. *Tableau.* [Online]
Available at: https://www.tableau.com/learn/articles/big-data-analytics
[Accessed 29 April 2022].

Taylor, D., 2022. *Guru99.* [Online]
Available at: https://www.guru99.com/what-is-big-data.html
[Accessed 29 April 2022].

Taylor, D., 2022. *Guru99.* [Online]
Available at: https://www.guru99.com/nosql-tutorial.html
[Accessed 29 April 2022].

Tom, 2021. *BRAND24.* [Online]
Available at: https://brand24.com/blog/sentiment-analysis/
[Accessed 29 April 2022].

tutorialspoint, 2022. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/mongodb/index.htm
[Accessed 29 April 2022].

Zhasa, M., 2021. *simiplilearn.* [Online]
Available at: https://www.simplilearn.com/tutorials/hadoop-tutorial/what-is-hadoop
[Accessed 29 April 2022].