Annexure-1 (A typical Specimen of Title Page)

ANIME RECOMMENDATION SYSTEM

A Project Work

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN AI AND ML

Submitted by:

SOURAV KUMAR SINGH

20BCS6685

AMANJOT SINGH

20BCS6702

AMISHA KHANNA

20BCS6712

RIVI VIG

20BCS6724

Under the Supervision of:

Miss LEEZA SHARMA

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

MAY, 2022

Annexure-2

DECLARATION

We, **'Sourav Kumar Singh', 'Amanjot Singh', 'Amisha Khanna', 'Rivi Vig'**, students of **'Bachelor of Engineering in CSE with specialization in AIML'**, **session: 2021-2022**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**Anime Recommendation Generator'** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Sourav Kumar Singh
Candidate UID: 20BCS6685

Amanjot Singh
Candidate UID: 20BCS6702

Amisha Khanna
Candidate UID: 20BCS6712

Rivi Vig
Candidate UID: 20BCS6724

Date: 19.05.2022

Place: Chandigarh University, Gharuan, Punjab

ABSTRACT

Entertainment is the action of being provided with amusement or enjoyment. It is an escape from the real world. It distracts you from stress causing factors in your life helps you relax. It helps in keeping your mind engaged while there is nothing else to do. During the pandemic, when everyone was forced to stay in their homes, people found diverse ways to engage themselves. Watching movies and series especially on OTT platforms became a favorite way to pass their time for many. People started experimenting with different genres and discovered a lot of new stuff. Many people discovered and rediscovered their love for anime.

Anime refers to animation produced specifically is Japan. Japan has been producing anime for over a century now with the first ever anime being produced in 1917, named ‘Katsudo Shashin’ by and unknown creator. Since then, thousands of anime have been produced ranging from movies to series. This ever-growing industry, presently, has a global market size of roughly USD 26 billion and is expected to grow even further in the coming years. Anime is categorized into various genres including shojo and shonon. Anime like Demon Slayer, Naruto, Attack On Titan and Fruits Basket are very popular among the global audience.

With so much anime content to watch, one often gets confused as to what to watch. According to a study, the average American wastes approximately forty-eight hours per year deciding what to watch. To solve this issue, we have come up with an anime recommendation system. Using this, one can get an anime recommendation in one click. Another motive of an anime recommendations generator is to promote anime in India and to remove this preconceived notion that ‘anime is just for kids.

This recommendation system can also help an individual or a company to add/view the highest rated amine. In this, we will conduct exploratory data analysis to familiarize the reader with the data presented. We will create a baseline model using Singular Valuable Decomposition (SDV). We will then make memory-based models that will look at user-based v item based.

ACKNOWLEDGEMENT

We are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete.
We would like to express our special thanks of gratitude to our teacher, “Miss Leeza Sharma” who gave us the golden opportunity to do this wonderful project on the topic "Anime Recommendations Generator", which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.
Any attempt at any level can 't be satisfactorily completed without the support and guidance of our parents and friends.

Thanking you,
Team 7

Annexure-3 **(A typical specimen of table of contents)**

Table of Contents

List of Tables

.

List of Figures

Figure Title ***page***

List of Symbols

Symbol	*Description*
A_{st}	*Area of steel reinforcement bars on tension face*
A_{sc}	*Area Of steel reinforcement bars on compression face Area of two*
A_{sv}	*legs of the closed stirrups*
b	*Breadth of rectangular beam section*
d	*Effective depth of rectangular beam section*
d'	*Effective cover on compression face*
$f_{c,ave}$	*Average compressive stress in concrete*
f_{sc}	*Stress in steel on the compression side*
f_y	*Characteristic strength of steel reinforcement bars Spacing of the*
S_v	*stirrups*
x_u	*Depth of neutral axis from compression face*
$\bar{x}$	*Depth of centroid of the compression block in concrete*
τ_c	*Shear strength offered by concrete*

1 INTRODUCTION

The word ‘anime’ is derived from the English word ‘animation’ and it refers to hand-drawn and computer-generated animation originating from Japan. Globally, anime refers to animation produced specifically in Japan, whereas, in Japan, it refers to all type of animation irrespective of its origin. Anime in Japan began in early 20th century. The earliest Japanese animation is claimed to be Katsudo Shashin, created circa 1907 by and unknown creator.

A frame from Katsudo Shashin

In 1917, the first professional and publicly displayed works began to appear; animators such as Oten Shimokawa, Seitaro Kitayama and Junichi Kouchi produced numerous films, the oldest surviving of which is Kouchi’s Namakura Gatana.

A frame from Namakura Gatana

The first talkie anime was Chikara to Onna no Yo no Naka (1933), a short film produced by Masaoka. The first feature-length anime film was Momotaro: Sacred Sailors (1945), produced by Seo with a sponsorship from Imperial Japanese Navy. The 1950s saw a proliferation of short, animated advertisements created for television.

In the 1990s, anime also began to attract the Western audience with successes like Dragon Ball Z, which was dubbed into more than a dozen languages worldwide.

Dragon Ball Z

Demon Slayer: Kimetsu no Yaiba the Movie: Mugan Train became one of the world's highest grossing films of 2020. It made 10 billion yen in 10 days.

A frame from Demon Slayer: Mugan Train

Japanese animation studios emphasize more on art quality, unlike Disney animation, which focuses more on movement. Anime scenes focuses on achieving three dimensional views, and backgrounds are instrumental in creating the atmosphere of the work. The cinematic effects of anime differentiates itself from the stage plays found in American animation. Anime is cinematically shot as if by camera. In anime, the animation is produced before the voice acting, contrary to American animation which does the voice acting first.

Anime are often categorized into different genres including children's (kodomo), girls' (shojo) and boys' (shonen). Some popular kodomo anime include Pokemon and Doraemon.

A frame from Pokemon

Some famous shojo anime are Orange and Fruits Basket.

A frame from Fruits Basket

Naruto, Attack on Titan, Demon Slayer and Jujutsu Kaisen are some of the most popular anime of shonen genre.

The anime industry consists of more than 430 production companies. Thousands of anime have been produced till date. With so much diverse anime content available to watch, it creates confusion for a person as to what to watch. To solve this problem, we have come up with an anime recommendation generator.

A recommendation generator generates recommendation on a single click. It uses a large dataset and gives a random recommendation. Here, the dataset used is that on anime.

1.1 PROBLEM DEFINITION

Deciding what to watch might be one of the most dilemmatic decisions one has to make. Especially, if more than one person is going to decide. People often end up fighting while deciding what to watch. With so much content from across various genres available to us in just one click, it is a real headache. Deciding on what to watch consumes so much time that sometimes we are left with no time to watch by the time we decide what to watch. Anime is one of the fastest growing sections of entertainment industry today. It caters to us with content on a variety of genres. It is being loved by people from all sections of society. 6 6 | P a g e So, to end the conundrum of what to watch, an anime recommendations generator is a great solution. On just one click, a recommendation will be generated and in case you have already watched the show then just click once again to generate another recommendation. Another motive of an anime recommendations generator is to promote anime in India and to remove this preconceived notion that ‘anime is just for kids. This recommendation system can also help an individual or a company to add/view the highest rated amine.

1.2 PROJECT OVERVIEW/SPECIFICATIONS

INTRODUCTION

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer)system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With an increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements

If your tasks are small and can fit in a complex sequential processing, you don't need a big system. You could even skip the GPUs altogether. A CPU such as i7–7500U can train an average of ~115 examples/second. So, if you are planning to work on other ML areas or algorithms, a GPU is not necessary. This works for our case

If your task is a bit intensive, and has a manageable data, a reasonably powerful GPU would be a better choice for you. A laptop with a dedicated graphics card of high end should do the work. There are a few high end (and expectedly heavy) laptops like Nvidia GTX 1080 (8 GB VRAM), which can train an average of ~14k examples/second. In addition, you can build your own PC with a reasonable CPU and a powerful GPU, but keep in mind that the CPU must not bottleneck the GPU. For instance, an i7-7500U will work flawlessly with a GTX 1080 GPU.

`

2 LITERATURE REVIEW

Kim et al. [25] proposed VUDDY, which is a scalable approach for detection of vulnerable code clones. This approach can detect vulnerabilities efficiently and accurately in large software. They able to achieve extreme level of scalability by using function-level granularity and a length-filtering techniques that decreases number of signature comparisons. Most interesting feature of this technique is that it can even detect variants of known vulnerabilities. To achieve extreme level of scalability, they used function-level granularity and length-filtering techniques to reduce number of signature comparisons.

Artificial Intelligence (AI) can be defined as the study of programmed systems that can simulate human intelligence and activities, such as perceiving, thinking, learning, and acting. Examples of these tasks are visual perception, speech recognition, decision-making, and translation between languages. Artificial intelligence is already part of our everyday lives: self driving cars, navigation systems, chatbots, smart non-player characters. AI has developed as a field of research but also as a technology that expands across a wide range of applications. What differentiates AI. from other digital technologies is that AIs are set to learn from their environments in order to take autonomous decisions

2.1 Literature Review Summary

Table 2.1: Literature review summary

Year and citation	Article Title	Purpose of study	Tools / Software Used	Comparison of technique	Source (Journal / Conference)	Findings	Data set (if used)	Evaluation parameters
2010								

3 PROBLEM FORMULATION

During software development, clones can occur in software intentionally or unintentionally. Developers tend to clone fragments of software during development to save efforts and expedite the development process. .....

From the literature review, it is observed that studies highlight the need of efficient and scalable approach for detecting code clones having software vulnerability. The existing techniques are not able to detect all types of vulnerable code clones. Different approaches suffer from high false negative rate and not scalable to large software systems due to high time complexity. So firstly, there is a need..... Second same subject systems should be used to compare the approaches which detect.

4 RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of an approach for anime recommendation system. The proposed aim will be achieved by dividing the work into following objectives:

1. To understand and explore various types of software vulnerabilities existing in recommendation systems.
2. To study and analyze various recommendation techniques that are suitable for anime recommendation system.
3. To design and develop the technique for anime recommendation system.
4. To verify and validate the proposed system.
5. To help the company/individual to add/view the highest rated anime.
6. To help people get anime recommendations.
7. To change the notion that anime is just for kids.

5 METHODOLOGY

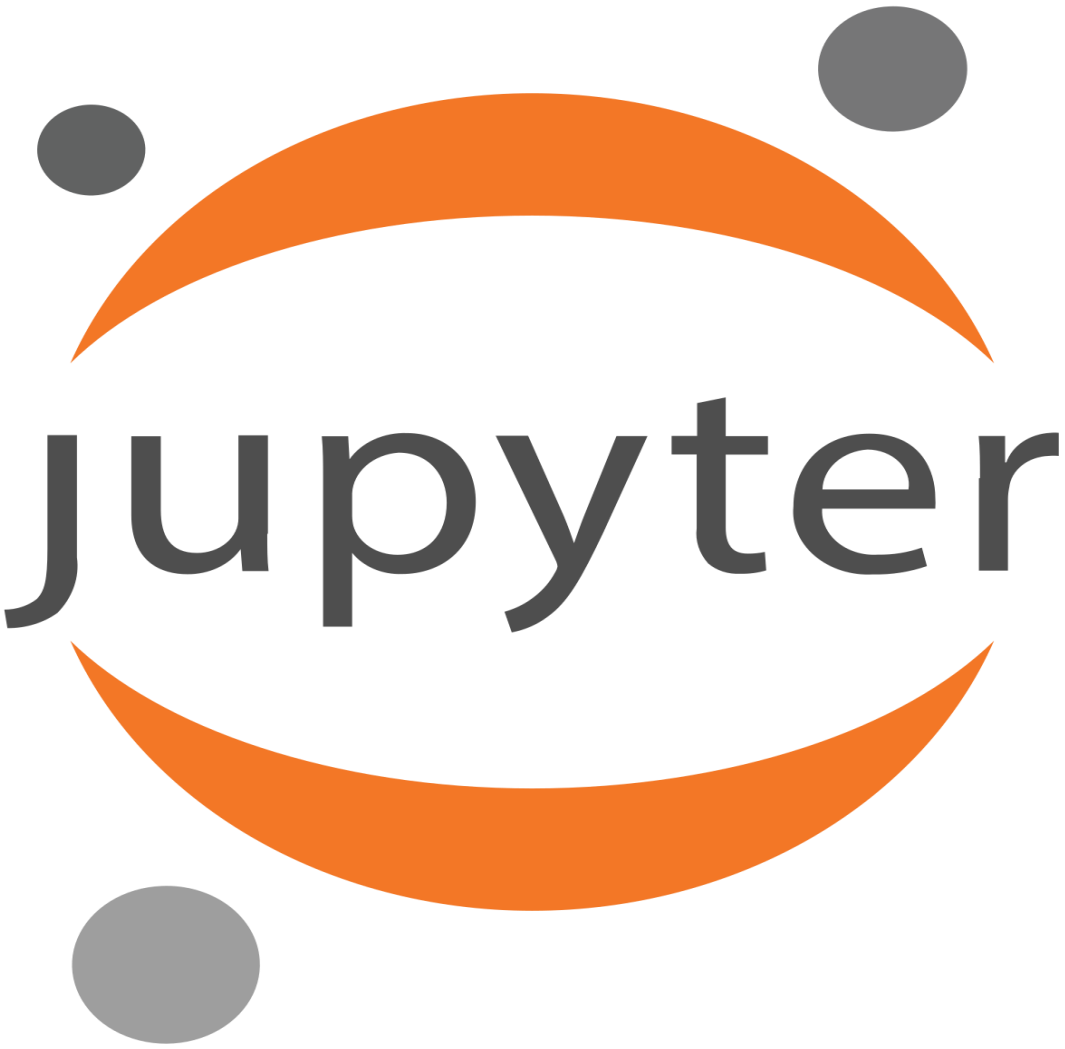
This research employed quantitative research method. Quiet some respondents participated through convenience sampling for this research. Respondents are university students from our university. As for the instrument, questionnaire was developed in conducting this research. The items in the questionnaire include demographic and impacts (negative and positive) of anime. Questionnaire was self-developed by the researchers thus; pilot test was conducted to check reliability of the instrument. Pilot test revealed a reliability score. Many of which suggested that the questionnaire is valid to be used. Analysis of data includes comparison between two independent variables; female and male university students. Independent T- test will be used to analyze the comparison.

SOFTWARE REQUIREMENTS

Jupyter Notebook:

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.



Google Collab:

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

What is the difference between Jupyter and Colab?

Jupyter is the open source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything.

Data set:

This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings.

Anime.csv

anime_id - myanimelist.net's unique id identifying an anime.

name - full name of anime.

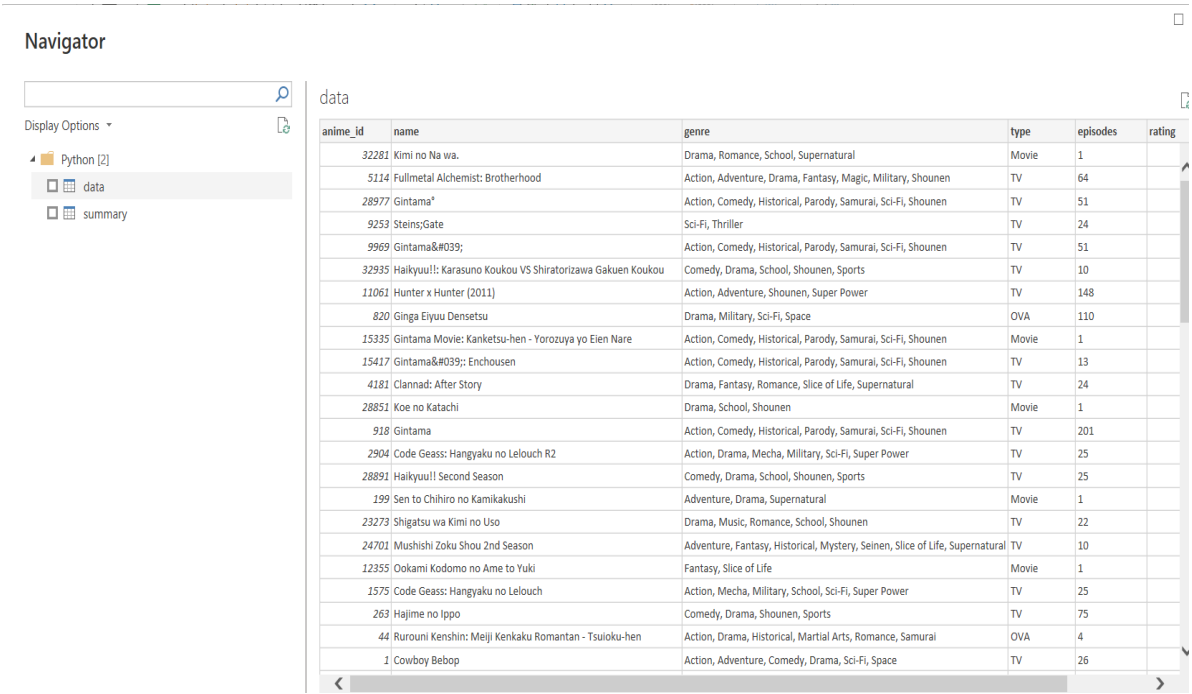
genre - comma separated list of genres for this anime.

type - movie, TV, OVA, etc.

episodes - how many episodes in this show. (1 if movie).

rating - average rating out of 10 for this anime.

members - number of community members that are in this anime's "group".



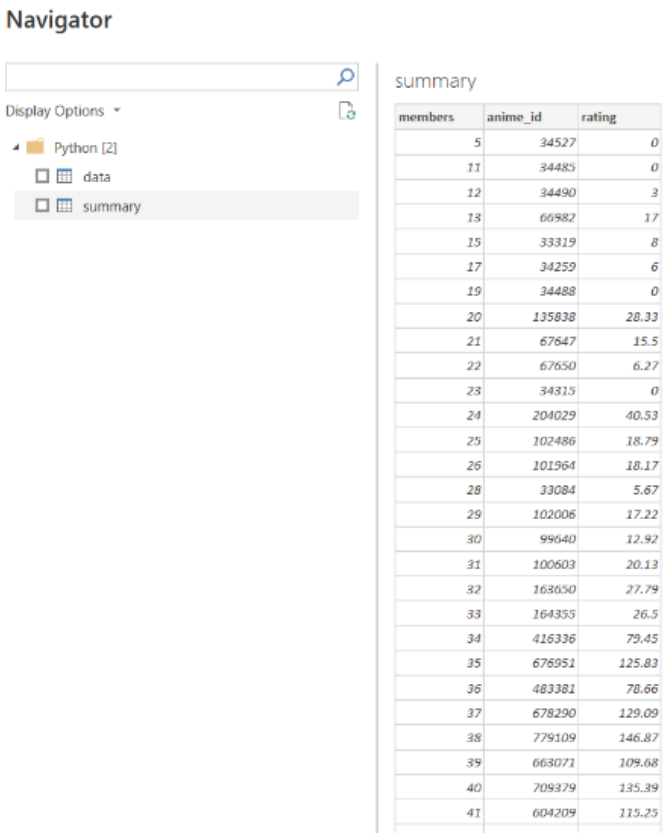
anime_id	name	genre	type	episodes	rating
32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	
5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Military, Shounen	TV	64	
28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	51	
9253	Steins;Gate	Sci-Fi, Thriller	TV	24	
9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	51	
32935	Haikyuu!!: Karasuno Koukou VS Shiratorizawa Gakuen Koukou	Comedy, Drama, School, Shounen, Sports	TV	10	
11061	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	TV	148	
820	Ginga Eiyuu Densetsu	Drama, Military, Sci-Fi, Space	OVA	110	
15335	Gintama Movie: Kanketsu-hen - Yorozuya yo Eien Nare	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	Movie	1	
15417	Gintama': Enchousen	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	13	
4181	Clannad: After Story	Drama, Fantasy, Romance, Slice of Life, Supernatural	TV	24	
28851	Koe no Katachi	Drama, School, Shounen	Movie	1	
918	Gintama	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	201	
2904	Code Geass: Hangyaku no Lelouch R2	Action, Drama, Mecha, Military, Sci-Fi, Super Power	TV	25	
28891	Haikyuu!! Second Season	Comedy, Drama, School, Shounen, Sports	TV	25	
199	Sen to Chihiro no Kamikakushi	Adventure, Drama, Supernatural	Movie	1	
23273	Shigatsu wa Kimi no Uso	Drama, Music, Romance, School, Shounen	TV	22	
24701	Mushishi Zoku Shou 2nd Season	Adventure, Fantasy, Historical, Mystery, Seinen, Slice of Life, Supernatural	TV	10	
12355	Ookami Kodomo no Ame to Yuki	Fantasy, Slice of Life	Movie	1	
1575	Code Geass: Hangyaku no Lelouch	Action, Mecha, Military, School, Sci-Fi, Super Power	TV	25	
263	Hajime no Ippo	Comedy, Drama, Shounen, Sports	TV	75	
44	Rurouni Kenshin: Meiji Kenkaku Romantan - Tsuioku-hen	Action, Drama, Historical, Martial Arts, Romance, Samurai	OVA	4	
1	Cowboy Bebop	Action, Adventure, Comedy, Drama, Sci-Fi, Space	TV	26	

Rating.csv

user_id - non identifiable randomly generated user id.

anime_id - the anime that this user has rated.

rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).



members	anime_id	rating
5	34527	0
11	34485	0
12	34490	3
13	66982	17
15	33319	8
17	34259	6
19	34488	0
20	135838	28.33
21	67647	15.5
22	67650	6.27
23	34315	0
24	204029	40.53
25	102486	18.79
26	101964	18.17
28	33084	5.67
29	102006	17.22
30	99640	12.92
31	100603	20.13
32	163650	27.79
33	164355	26.5
34	416336	79.45
35	676951	125.83
36	483381	78.66
37	678290	129.09
38	779109	146.87
39	663071	109.68
40	709379	135.39
41	604209	115.25

Numpy:

which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. It stands for 'Numerical Python'.

Pandas:

is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

Sklearn:

is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction

Matploatlib:

is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Scipy:

is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

Surprize:

is a Python scikit for building and analyzing recommender systems that deal with explicit rating data. Surprise was designed with the following purposes in mind: Give users perfect control over their experiments.

Random:

module that can generate random numbers. These are pseudo-random number as the sequence of number generated depends on the seed. If the seeding value is same, the sequence will be the same

6 RESULTS AND DISCUSSION

We have completed 40% of our work, includes coding and research part. we have downloaded anime and user ratings from https://www.kaggle.com/CooperUnion/anime-recommendations-database .

we will conduct exploratory data analysis to familiarise the reader and ourself with the data presented. From there we have created a baseline model using Singular Value Decomposition(SVD).

we will conduct Analysis, then do memory-based models that will look at user-based v item based. We will use KNNBase, KNNBaseline, and KNNWithMeans.

Project:

Below, we can see step-by-step guide to a recommendation system for anime.

Import:

```
import pandas as pd
import numpy as np

import random from random import randint

import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
from scipy.sparse import csc_matrix
from scipy.sparse.linalg import svds

from surprise.model_selection import train_test_split
from surprise.model_selection import GridSearchCV
from surprise.model_selection import cross_validate

from surprise.prediction_algorithms import KNNWithMeans, KNNBasic,
KNNBaseline
from surprise.prediction_algorithms import knns
from surprise.prediction_algorithms import SVD
from surprise.similarities import cosine, msd, pearson
```

```
from surprise import accuracy
from surprise import Reader
from surprise import Dataset
```

For most of this project, we decided to stick with the above and it's worked nicely. We have tried additional models below in google colab, we will add the necessary code for the import library there.

Scrub/Clean:

```
anime_df = pd.read_csv('./anime.csv')
anime_df.head()
anime_df.shape
```

This is important to use shape because it helps with seeing any null values.

```
anime_df.dropna(inplace=True)
Dropping null Values.
```

It's important to show what each column represents when cleaning so that we have an idea of what can be scrubbed, transformed, or if we need to conduct some feature engineering.

Analysis:

First thing's first! We did a base model. We have started with transforming the data.

```
data = sample[['user_id', 'anime_id', 'rating_x']]

Reader(line_format='user item rating', sep='')
anime_loaded_data=Dataset.load_from_df(data,reader)

#train_test_split

trainset, testset = train_test_split(anime_loaded_data,
test_size=.2)

anime_loaded_data
```

Next, we'll instantiate.

```
#INSTANTIATE the SVD and fit only the train set
svd = SVD()
svd.fit(trainset)
```

Now to look at predictions and accuracy. This is important for comparing your learner models.

```
predictions = svd.test(testset) #
accuracy.rmse(predictions)
```

This is my baseline: RMSE: 2.3128, 2.3127

Well, it’s definitely not between 0 and 1. These scores really depend on the domain.

Let’s reduce the error

```
params = {'n_factors': [20,50,100],
'reg_all': [.02,.05, .10]}

gridsearch_svd1 = GridSearchCV(SVD, param_grid=params, n_jobs=-
1,joblib_verbose=3)

gridsearch_svd1.fit(anime_loaded_data)

print(gridsearch_svd1.best_score)

print(gridsearch_svd1.best_params)
```

{‘rmse’: 2.3178, ‘mae’: 2.2080} {‘rmse’: {‘n_factors’: 20, ‘reg_all’: 0.02}, ‘mae’: {‘n_factors’: 20, ‘reg_all’: 0.02}}

It’s increased, Lets try another model.

K-Nearest Neighbors (KNN) Basic Algorithm:

```
knn_basic = KNNBasic(sim_options={'name':'pearson',
'user_based':True}, verbose=True)
cv_knn_basic = cross_validate(knn_basic, anime_loaded_data,
n_jobs=2)

for i in cv_knn_basic.items():
      print(i)
     print('-----------------')
     print(np.mean(cv_knn_basic['test_rmse']))
```

We got output of 2.3178203641229667

We can do the same for the mean squared distance(msd).

```
knn_basic_msd = KNNBasic(sim_options = {'name': 'msd', 'user-
based':True})
cv_knn_basic_msd = cross_validate(knn_basic_msd, anime_loaded_data,
n_jobs=2)

for i in cv_knn_basic_msd.items():
print(i)
print('-----------------')
print(np.mean(cv_knn_basic_msd['test_rmse']))
```

Output: 2.31787540672289

KNNBaseline:

```
knn_baseline = KNNBaseline(sim_options={'name': 'pearson',
```

```
'user_based':True})

cv_knn_baseline = cross_validate(knn_baseline,
anime_loaded_data, n_jobs=3)
for i in cv_knn_baseline.items():
      print(i)
     print('------------------')
     print(np.mean(cv_knn_baseline['test_rmse']))
```

Ouput: 2.317895626569356

KNN Baseline with pearson_baseline:

```
knn_pearson_baseline = KNNBaseline(sim_options={'name':
'pearson_baseline', 'user_based':True})

cv_knn_pearson_baseline = cross_validate(knn_pearson_baseline,
anime_loaded_data, n_jobs=3)

for i in cv_knn_pearson_baseline.items():
print(i)
print('--------------------')
print(np.mean(cv_knn_pearson_baseline['test_rmse']))
```

Output: -2.317801933128254

KNNWithMeans:

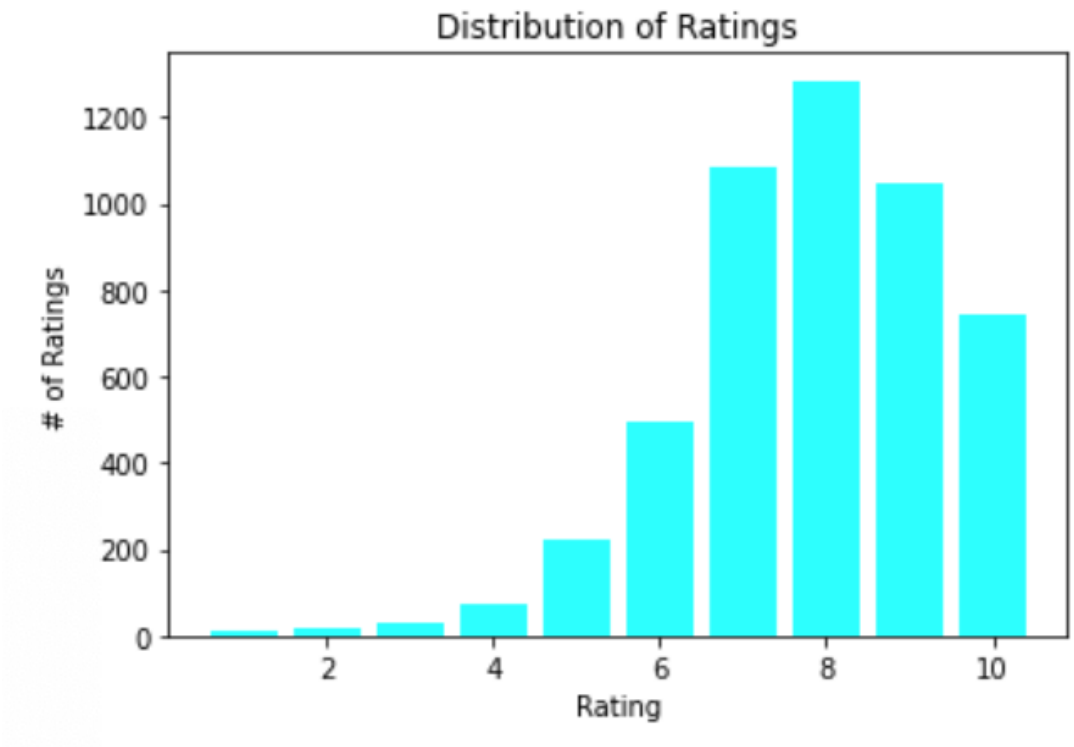
```
knn_means = KNNWithMeans(sim_options={'name': 'pearson',
'user_based': True})
cv_knn_means = cross_validate(knn_means, anime_loaded_data,
n_jobs=3)

for i in cv_knn_means.items():
print(i)
print('------------')
print(np.mean(cv_knn_means['test_rmse']))
```

Output: 2.3185632763331805

Exploratory Data Analysis (EDA):

Distribution of User Ratings

```
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()
```




```
print("Number of Users:", df_merge.user_id.nunique()# print("Average Number of
Reviews per User:", df_merge.shape[0])/df_merge.user_id.nunique()avg_rate_peruser
= df_merge.shape[0]user = df_merge.user_id.nunique()
avg_num_review_per_user = avg_rate_peruser/userprint("Average Number of Reveiws
per User:", avg_num_review_per_user)
```

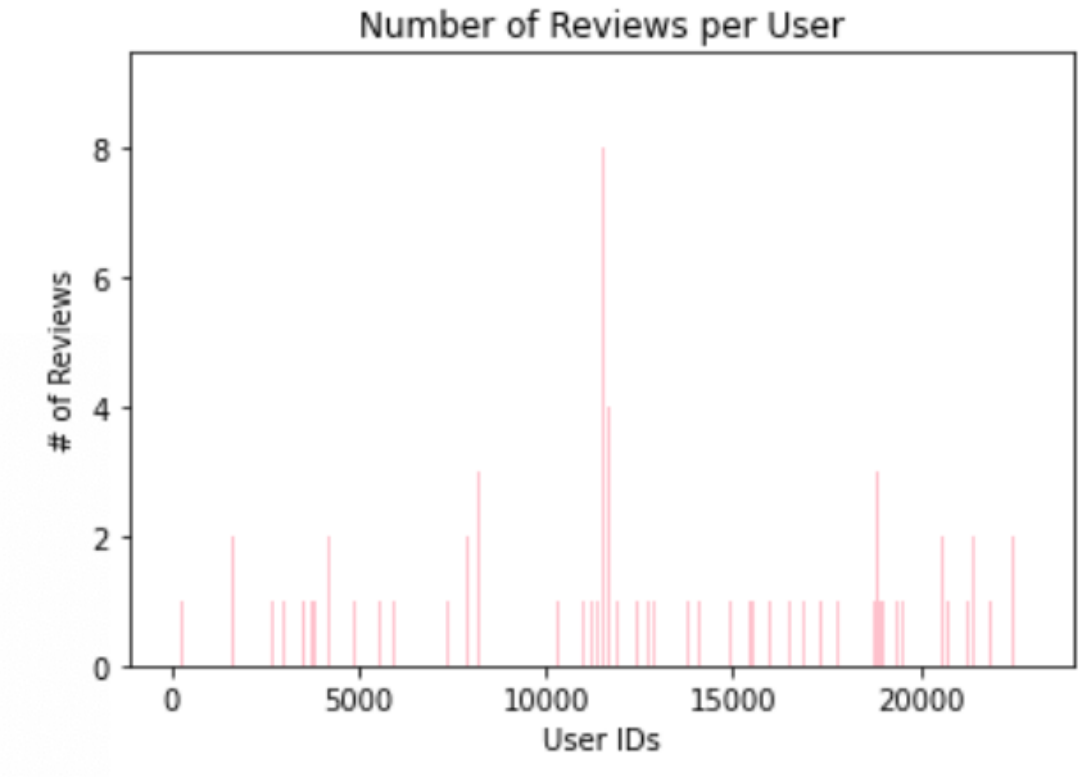
Number of Users: 15382

Average Number of Reviews per User: 88.69

```
sample['user_id'].value_counts()
```

The sample size of 5000, gives us a total of 3,381 users who have done reviews. The above is on the entire set.

Number of Reviews Per User



It seems that from our sample set, a lot of the scores given are 1s',2 and 4s'.

Different Types of Anime

```
print("Number of users:", sample.user_id.nunique())
print("Number of types of different anime:", sample.type.nunique())
print("Types of type:", sample.type.value_counts())
```

We have 6 different types of anime.

TV 3492

Movie 666

OVA 461

Special 314

ONA 46

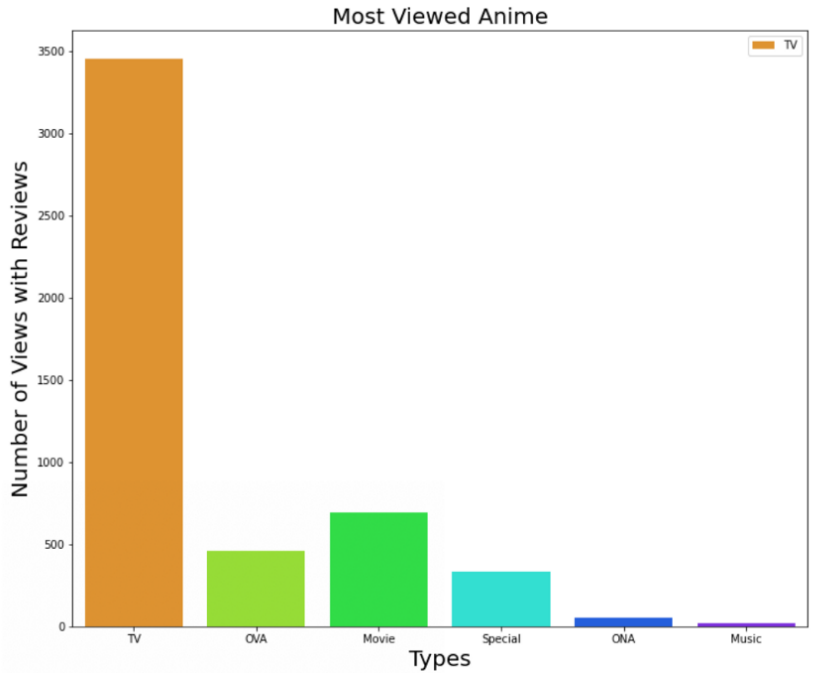
Music 21

Name: type, dtype: int64

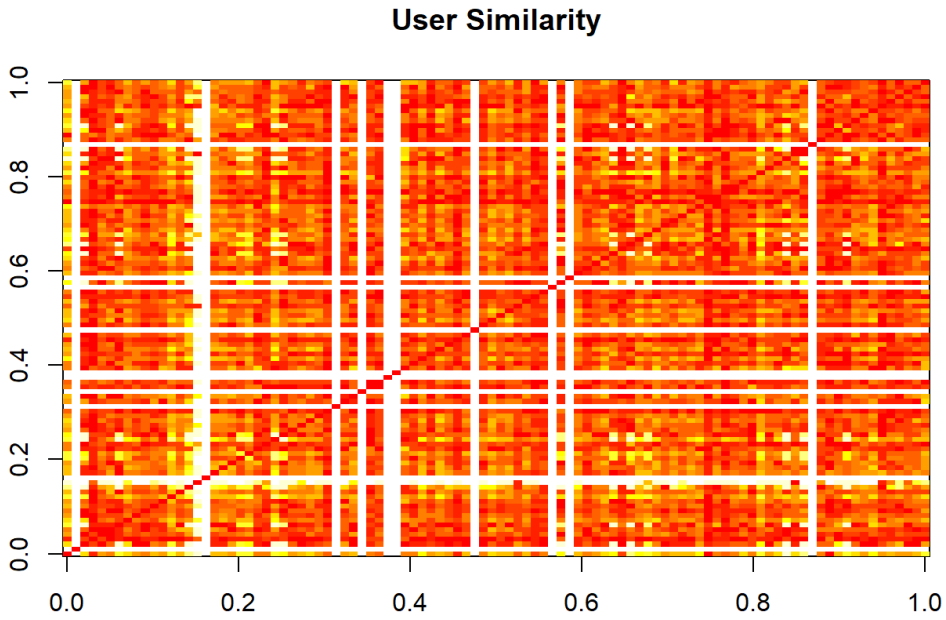
Most Viewed Anime

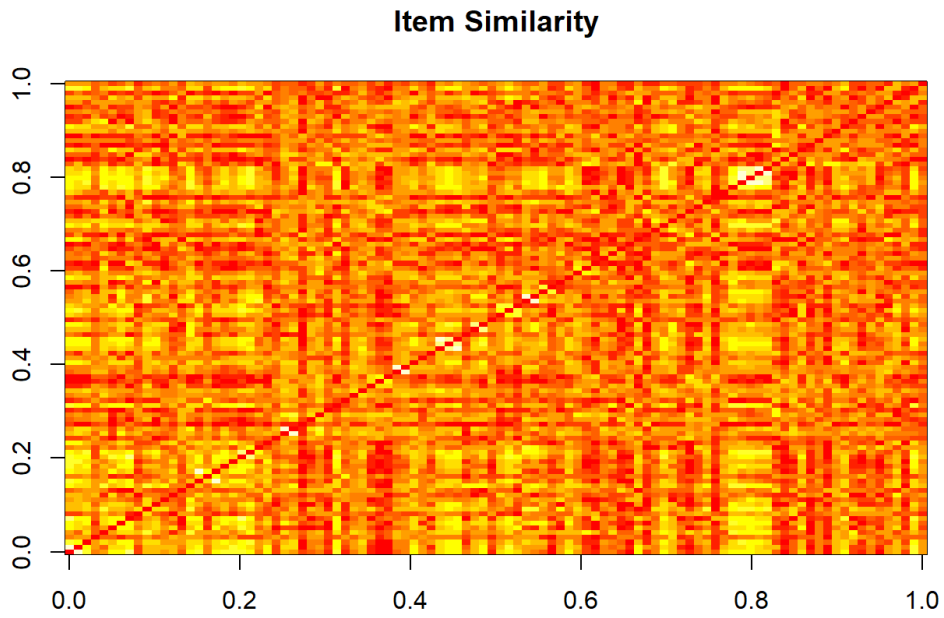
```
fig = plt.figure(figsize=(12,10))
sns.countplot(sample['type'], palette='gist_rainbow')
plt.title("Most Viewed Anime", fontsize=20)
plt.xlabel("Types", fontsize=20)
plt.ylabel("Number of Views with Reviews", fontsize = 20)
```

```
plt.legend(sample['type'])
plt.show()
```



Similarity:





Evaluation:

Recommender	TopN Accuracy							
	TP	FP	FN	TN	precision	recall	TPR	FPR
anime_item_acc1	0.3035230	7.967480	114.7751	1592.954	0.0385451	0.0035823	0.0035823	0.0047436
anime_svd_acc1	1.4336043	8.566396	113.6450	1592.355	0.1433604	0.0156471	0.0156471	0.0051708
anime_hy_acc1	0.9403794	9.059621	114.1382	1591.862	0.0940379	0.0053911	0.0053911	0.0060231

Recommender	Ratings Accuracy		
	RMSE	MSE	MAE
anime_item_acc2	1.596660	2.549324	1.107453
anime_svd_acc2	1.688743	2.851854	1.235027
anime_hy_acc2	1.583035	2.506001	1.124243

7 Future Scope:

In future, a website can be made with a front end or some application.

8 REFERENCES

[1] D. Rattan, R. Bhatia, and M. Singh, "Software clone detection: A systematic review," *Information and Software Technology*, vol. 55, no. 7, pp. 1165–1199, Jul. 2013.

[2] J. F. Islam, M. Mondal, and C. K. Roy, "Bug Replication in Code Clones: An Empirical Study," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 68–78.

[3] M. R. Islam and M. F. Zibran, "A Comparative Study on Vulnerabilities in Categories of Clones and Non-cloned Code," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 8–14.

[4] M. R. Islam, M. F. Zibran, and A. Nagpal, "Security Vulnerabilities in Categories of Clones and Non-Cloned Code: An Empirical Study," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017, pp. 20–29.

[5] C. K. Roy, M. F. Zibran, and R. Koschke, "The vision of software clone management: Past, present, and future (Keynote paper)," in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 18–33.

[6] J. Krinke, "A Study of Consistent and Inconsistent Changes to Code Clones," in *14th Working Conference on Reverse Engineering (WCRE 2007)*, 2007, pp. 170–178.

[7] D. Chatterji, J. C. Carver, N. A. Kraft, and J. Harder, "Effects of cloned code on software maintainability: A replicated developer study," in *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013, pp. 112–121.

[8] D. Rattan, R. Bhatia, and M. Singh, "An Empirical Study of Clone Detection in MATLAB/ Simulink Models," *International Journal of Information and Communication Technology*.

[9] D. Rattan, R. Bhatia, and M. Singh, "Detecting High Level Similarities in Source Code and Beyond," *International Journal of Energy, Information and Communications*, vol. 6, no. 2, pp. 1–16, 2015.

[10] D. Rattan, R. Bhatia, and M. Singh, "Detection and Analysis of Clones in UML Class Models," *International Journal of Software Engineering, IJSE*, vol. 8, no. 2, pp. 66–99, 2015.

[11] D. Rattan, R. Bhatia, and M. Singh, "Model clone detection based on tree comparison," in *2012 Annual IEEE India Conference (INDICON)*, 2012, pp. 1041–1046.

[12] C. K. Roy and J. R. Cordy, "NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization," in *2008 16th IEEE International Conference on Program Comprehension*, 2008, pp. 172–181.

[13] M. Mondal, C. K. Roy, and K. A. Schneider, "SPCP-Miner: A tool for mining code clones that are important for refactoring or tracking," in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2015, pp. 484–488.