National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Varshini Subburaj
Student ID: x22153977

School of Computing
National College of Ireland

Supervisor: Dr Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Varshini Subburaj
Student ID:	x22153977
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr Hicham Rifai
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	479
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Varshini Subburaj
x22153977

1 Introduction

The present document contains all the necessary information required to reproduce the results of the research project named "Deep Anime Recommendation System: Recommending Anime Using Hybrid Filtering." Furthermore, the paper provides a comprehensive description of the basic system requirements that are necessary for the effective implementation and functioning of the proposed anime recommendation system.

2 System Specification

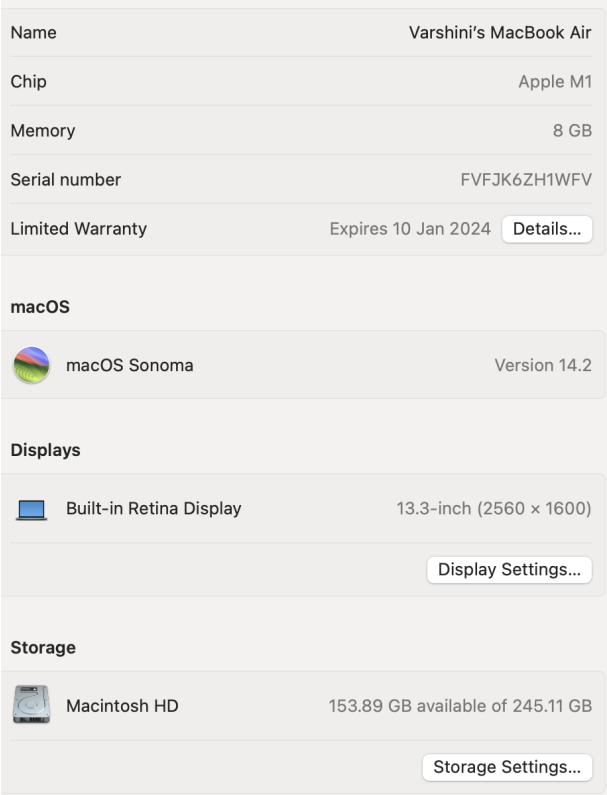

Figure 1: System Specification

This system is outfitted with the Apple M1 processor, which offers sophisticated processing capabilities. The system guarantees effective multitasking and smooth execution

of resource-intensive activities with its 8 GB of RAM. The device's unique identifier (FVFJK6ZH1WFV) functions as a separate marker for system identification.

The system operates on macOS Sonoma, Version 14.2, and makes use of the most recent features and optimisations provided by Apple's operating system. The generous storage capacity of 246 GB effectively meets the data storage needs, guaranteeing appropriate room for the information and models used in the recommendation system.

This system specification is designed to facilitate the effective construction and operation of the Deep Learning-Based Anime Recommendation System by offering the required computing resources and interoperability with various platforms.

3 Software requirement

The Table 1 displays the essential minimal software requirements needed to execute this project.

Table 1: Feature Description

Feature	**Description**
Language	Python
IDE	Jupyter Notebook 6.5.2 using Terminal
Data Preprocessing and EDA	Pandas, NumPy, Matplotlib, Seaborn, Plotly, scikit-learn libraries
Visualization	Matplotlib and Seaborn libraries

4 Installing Environment and Downloading the Required Files

Using Terminal to Open the Jupyter Notebook" section, the document outlines the process of acquiring and setting up the necessary development environment for the Deep Anime Recommendation System.

The datasets[1] utilised include the Anime Information Dataset (anime.csv), which contains comprehensive information about anime such as title, rating, genres, and format; the User Ratings Dataset (animelist.csv), which records individual user ratings and viewing status for anime; and the Anime Synopsis Dataset (anime with synopsis.csv), which enhances the dataset by including synopses for content-based recommendation.

5 Implementation and Evaluation

The code starts by importing and examining anime datasets, handling missing values, and doing data cleansing. The subsequent analysis involves Exploratory Data Analysis (EDA) and visualisations, including diverse elements such as highest-rated anime, popularity, and genres.

A subset of user ratings is collected to accommodate memory limitations. An integrated recommendation system is deployed, merging content-based and collaborative filtering techniques. The Surprise library is implemented for the purpose of collaborative

[1]Dataset: `https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020`

filtering, and the performance of the model is assessed using a separate test set. The hybrid technique is used to provide personalised suggestions for both users and anime titles.

Importing Libraries

In [1]:
```
import os
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
import plotly.graph_objects as go
```

Data Preprocessing

In [2]:
```
anime_info = pd.read_csv('/Users/varshini/Downloads/RIC_Dataset/anime.csv')
anime_list = pd.read_csv('/Users/varshini/Downloads/RIC_Dataset/animelist.csv')
anime_synop = pd.read_csv('/Users/varshini/Downloads/RIC_Dataset/anime_with_synopsis.csv')
```

Figure 2: Data Loading and Exploration

	MAL_ID	Popularity	Members	Favorites	Watching	Completed	On-Hold	Dropped	Plan to Watch
count	17562.000000	17562.000000	1.756200e+04	17562.000000	17562.000000	1.756200e+04	17562.000000	17562.000000	17562.000000
mean	21477.192347	8763.452340	3.465854e+04	457.746270	2231.487758	2.209557e+04	955.049653	1176.599533	8199.831227
std	14900.093170	5059.327278	1.252821e+05	4063.473313	14046.688133	9.100919e+04	4275.675096	4740.348653	23777.691963
min	1.000000	0.000000	1.000000e+00	0.000000	0.000000	0.000000e+00	0.000000	0.000000	1.000000
25%	5953.500000	4383.500000	3.360000e+02	0.000000	13.000000	1.110000e+02	6.000000	37.000000	112.000000
	scroll output; double click to hide		00 2.065000e+03	3.000000	73.000000	8.175000e+02	45.000000	77.000000	752.500000
75%	35624.750000	13145.000000	1.322325e+04	31.000000	522.000000	6.478000e+03	291.750000	271.000000	4135.500000
max	48492.000000	17565.000000	2.589552e+06	183914.000000	887333.000000	2.182587e+06	187919.000000	174710.000000	425531.000000

Figure 3: Data Preprocessing

EDA and Visualization

```
edacol = ['anime_id', 'Name','English name', 'Score', 'Genres', 'Type', 'Aired', 'Premiered','Rating','Source','Epis
eda = anime_info[edacol]
eda.set_index('anime_id',inplace=True)
```

Top 5 animes based on Score/Rating

```
anime_info.sort_values('Score',ascending=False).head(5)
```

	anime_id	Name	Score	Genres	English name	Japanese name	Type	Episodes	Aired	Premiered	...	Score-10	Score-9	Score-8	Sc
3971	5114	Fullmetal Alchemist: Brotherhood	9.19	Action, Military, Adventure, Comedy, Drama, Ma...	Fullmetal Alchemist:Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	TV	64	Apr 5, 2009 to Jul 4, 2010	Spring 2009	...	714811.0	401507.0	199160.0	700
15926	40028	Shingeki no Kyojin: The Final Season	9.17	Action, Military, Mystery, Super Power, Drama,...	Attack on Titan Final Season	進撃の巨人 The Final Season	TV	16	Dec 7, 2020 to ?	Winter 2021	...	173154.0	63756.0	26016.0	87
5683	9253	Steins;Gate	9.11	Thriller, Sci-Fi	Steins;Gate	STEINS;GATE	TV	24	Apr 6, 2011 to Sep 14, 2011	Spring 2011	...	468504.0	275960.0	140914.0	577
14963	38524	Shingeki no Kyojin Season 3 Part 2	9.10	Action, Drama, Fantasy, Military, Mystery, Sho...	Attack on Titan Season 3 Part 2	進撃の巨人 Season3 Part.2	TV	10	Apr 29, 2019 to Jul 1, 2019	Spring 2019	...	327290.0	239451.0	110481.0	336

Figure 4: Feature Selection

```
top10_animerating=anime_ratingCount[['Name', 'rating']].sort_values(by = 'rating',ascending = False).head(10)
ax=sns.barplot(x="Name", y="rating", data=top10_animerating, palette="YlOrBr")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11, rotation=40, ha="right")
ax.set_title('Top 10 Anime based on rating counts',fontsize = 22)
ax.set_xlabel('Anime',fontsize = 20)
ax.set_ylabel('User Rating count', fontsize = 20)
```

Figure 5: Top 10 Anime based on rating counts

Content Filtering

```
anime_df['sypnopsis'] = anime_df['sypnopsis'].fillna('')
```

```
tfidf = TfidfVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0, stop_words='english')
tfidf_matrix = tfidf.fit_transform(anime_df['sypnopsis'])
tfidf_matrix.shape
```

```
(11091, 386042)
```

```
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
cosine_sim.shape
```

```
(11091, 11091)
```

```
anime_df = anime_df.reset_index()
titles = anime_df['Name']
indices = pd.Series(anime_df.index, index=anime_df['Name'])
```

```
def content_recommendations(title):
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:31]
    anime_indices = [i[0] for i in sim_scores]

    anime_lst = anime_df.iloc[anime_indices][['Name', 'Members', 'Score']]
    favorite_count = anime_lst[anime_lst['Members'].notnull()]['Members'].astype('int')
    score_avg = anime_lst[anime_lst['Score'].notnull()]['Score'].astype('float')
    C = score_avg.mean()
    m = favorite_count.quantile(0.60)
    qualified = anime_lst[(anime_lst['Members'] >= m) & (anime_lst['Members'].notnull()) & (anime_lst['Score'].notnu
    qualified['Members'] = qualified['Members'].astype('int')
    qualified['Score'] = qualified['Score'].astype('float')
    def weighted_rating(x):
        v = x['Members']
        R = x['Score']
        return (v/(v+m) * R) + (m/(m+v) * C)

    qualified['wr'] = qualified.apply(weighted_rating, axis=1)
    qualified = qualified.sort_values('wr', ascending=False).head(10)

    return qualified
```

Figure 6: Content Filtering

	Name	Members	Score	wr
1506	Naruto: Shippuuden	1543765	8.16	8.051279
824	Higurashi no Naku Koro ni	638491	7.95	7.762043
5998	The Last: Naruto the Movie	352160	7.76	7.528764
5623	Naruto: Shippuuden Movie 6 - Road to Ninja	223826	7.67	7.400756
7245	Boruto: Naruto the Movie	320603	7.50	7.342222
4381	Naruto: Shippuuden Movie 4 - The Lost Tower	172051	7.42	7.231966
2089	Naruto: Shippuuden Movie 1	211544	7.29	7.178705
3145	Naruto: Shippuuden Movie 2 - Kizuna	188680	7.29	7.171518
403	Naruto Movie 1: Dai Katsugeki!! Yuki Hime Shin...	215046	7.10	7.072324
826	Naruto Movie 2: Dai Gekitotsu! Maboroshi no Ch...	172509	6.88	6.956515

Figure 7: Content Filtering-recommendations

```
reader = Reader()
rating_data = Dataset.load_from_df(rating_df, reader)
svd = SVD()
```

```
trainset = rating_data.build_full_trainset()
```

```
svd.fit(trainset)
```

```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x177ea24a0>
```

```
svd.predict(1, 356, 5)
```

Figure 8: Collaborative Filtering

Hybrid Filtering

```
id_map = anime_df[['MAL_ID']]
id_map['id'] = list(range(1,anime_df.shape[0]+1,1))
id_map = id_map.merge(anime_df[['MAL_ID', 'Name']], on='MAL_ID').set_index('Name')
```

```
indices_map = id_map.set_index('id')
```

```
from sklearn.model_selection import train_test_split
# Assuming you have a DataFrame 'anime_df' with columns like 'MAL_ID', 'Name', 'Genres', 'Score'
# Replace these columns with your actual column names

# Step 1: Split the data into training and testing sets
train_data, test_data = train_test_split(anime_df, test_size=0.2, random_state=42)
```

Figure 9: Hybrid Filtering

	MAL_ID	Name	Genres	Score
213	245	Great Teacher Onizuka	Slice of Life, Comedy, Drama, School, Shounen	8.70
5022	10408	Hotarubi no Mori e	Drama, Romance, Shoujo, Supernatural	8.38
22	32	Neon Genesis Evangelion: The End of Evangelion	Sci-Fi, Dementia, Psychological, Drama, Mecha	8.51
9848	37675	Overlord III	Action, Magic, Fantasy, Game, Supernatural	7.95
28	47	Akira	Action, Military, Sci-Fi, Adventure, Horror, S...	8.17
7629	30484	Steins;Gate 0	Sci-Fi, Thriller	8.51
1077	1210	NHK ni Youkoso!	Comedy, Psychological, Drama, Romance	8.33
10616	40221	Kami no Tou	Action, Adventure, Mystery, Drama, Fantasy	7.66
8012	31859	Hai to Gensou no Grimgar	Action, Adventure, Drama, Fantasy	7.69
1573	1818	Claymore	Action, Adventure, Super Power, Demons, Supern...	7.78

Figure 10: Hybrid - Recommendation of Anime

```
# Evaluate the model on the test set
reader = Reader()
data = Dataset.load_from_df(test_data[['MAL_ID', 'Members', 'Score']], reader)
testset = data.build_full_trainset().build_testset()

predictions = model.svd.test(testset)
```

```
from surprise import accuracy
# Evaluate the model on the test set
reader = Reader()
data = Dataset.load_from_df(test_data[['MAL_ID', 'Members', 'Score']], reader)
testset = data.build_full_trainset().build_testset()

predictions = model.svd.test(testset)

# Compute and print RMSE, MSE, and MAE
rmse = accuracy.rmse(predictions)
mse = accuracy.mse(predictions)
mae = accuracy.mae(predictions)
```

```
RMSE: 1.8253
MSE: 3.3319
MAE:  1.6206
```

Figure 11: Evaluation Metrics