ANIME RECOMMENDATION SYSTEM

Project Report

Supervisor: Mr Mahesh Kumar Bhandari

Submitted By:

Raj Khatri AC-1235

Pratham Sharma AC-1232

2023

Department of Computer Science

ACHARYA NARENDRA DEV COLLEGE

ACKNOWLEDGEMENT

Raj Khatri and Pratham Sharma jointly undertaken data Mining project for their sixth semester, with Mr. Mahesh Kumar Bhandari providing guidance and supervision. The project required significant effort, but would not have been possible without the help and support of many individuals. The authors are grateful to Mr. Bhandari for his constant guidance, necessary information, and support throughout the project. They also extend their thanks and appreciation to their colleague and those who offered their abilities to help them complete the project.

Raj Khatri

Pratham Sharma

ACHARYA NARENDRA DEV COLLEGE

(University of Delhi)

CERTIFICATE

This is to certify that the project entitled, “ANIME RECOMMENDATION SYSTEM” has been done by Raj Khatri and Pratham Sharma in partial fulfillment of the requirements for the award of "Bachelor of Computer Science (Honors)" during Semester-VI at the "Acharya Narendra Dev College" under the supervision and guidance of Mr. Mahesh Kumar Bhandari.

Rajkhatri

Raj Khatri

Pratham Sharma

Mr. Mahesh Kumar Bhandari
(Supervisor)

Table of Contents

Topic **Page No**

Chapter 1
PROBLEM STATEMENT

With the increasing popularity of anime and the availability of a vast range of titles, it becomes challenging for users to identify relevant anime titles that match their preferences. Therefore, the development of an effective recommendation system can enhance user experience and increase user engagement. The project aims to use data mining techniques to analyze user behavior and anime features to predict and recommend relevant anime titles.

To develop an efficient recommendation system, the project requires a large dataset of user behavior and anime features. The dataset should be diverse, consisting of user ratings, reviews, and viewing history, as well as anime information such as genre, director, and studio. The data needs to be preprocessed, cleaned, and transformed to identify patterns and trends using data mining techniques such as K-Means clustering.

This project aims to develop an effective anime recommendation system that enhances user experience and increases user engagement by providing personalized and relevant anime title recommendations.

Chapter 2
DATA MINING TECHNIQUES

2.1. Data Mining Techniques

Data mining is the process of extracting valuable insights from large collections of data using automated methods. It forms a crucial aspect of knowledge discovery in databases, which involves transforming raw data into meaningful information. Data mining leverages various concepts, including artificial intelligence, machine learning, and pattern recognition, to perform tasks such as search algorithms and modeling techniques. It draws on ideas from these fields to identify patterns, correlations, and other meaningful information from data.

2.1.1. Classification

Classification ais a type of machine learning that falls under the category of supervised learning. Its purpose is to predict the classification or category of new observations by utilizing a training dataset. Essentially, it involves training the algorithm with labeled data to learn how to categorize new input data accurately. In simpler terms, it is a predictive modeling approach that assigns class labels to input data. An email spam detector is an example of a classification algorithm in action, where it can differentiate between spam and non-spam emails by using labeled training data.

2.1.2. Association

Association rule mining is a data mining technique that assists in identifying relationships between two or more items in a dataset. It aims to uncover hidden patterns or connections within large datasets. Association rules take the form of "if-then" expressions and support the possibility of interactions between data items in different types of databases.

In simpler terms, association rule mining helps to identify correlations between items in a dataset. For example, if a customer buys product A, they are likely to buy product B. This type of information is valuable for businesses, as it can help them identify opportunities for cross-selling or bundling products. Association rule mining is a powerful tool for uncovering relationships within data and is commonly used in market basket analysis, web usage mining, and recommendation systems.

2.1.3. Clustering

Clustering is the process of grouping abstract objects together based on their similarities. It involves the creation of classes or clusters of similar objects, which can help to identify patterns and structures within data. The goal of clustering is to divide a dataset into distinct groups or clusters, where the objects within each cluster are similar to each other, but different from those in other clusters.

In simpler terms, clustering is a technique that aims to identify similarities and differences within a dataset. For example, if we have a dataset of customer preferences, we can use clustering to group customers who have similar preferences or behaviors. This information can then be used to create targeted marketing campaigns or product recommendations for each group. Clustering is a commonly used technique in machine learning, data mining, and pattern recognition, and can be applied to a wide range of fields, including marketing, healthcare, and finance.

2.2. Data mining technique used for this project:

Clustering This project uses the data mining technique clustering**:**

2.2.1 K-Mean:

K-means is a clustering algorithm that aims to divide a dataset into k distinct clusters. It works by assigning each data point to the nearest cluster centroid, which is the average of all the data points in that cluster. The algorithm then iteratively updates the centroids by recalculating the mean of all the data points in each cluster. This process continues until the centroids no longer move, or until a maximum number of iterations is reached.

The K-means algorithm requires the user to specify the number of clusters, k, that they want to divide the dataset into. This can be a challenging task, as selecting the optimal k value requires some trial and error. One way to do this is to use the elbow method, which involves plotting the within-cluster sum of squares against the number of clusters and selecting the value of k where the change in sum of squares begins to level off.

K-means is a commonly used clustering algorithm that is widely applicable to a range of fields, including marketing, healthcare, and finance. It is relatively simple to implement and can handle large datasets efficiently. However, it is important to note that K-means has some limitations, such as its sensitivity to the initial placement of centroids and its assumption of isotropic clusters.

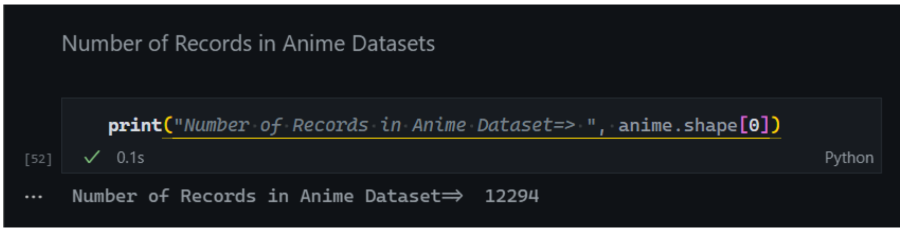
Chapter 3
Dataset Description

3.1 Dataset

This dataset includes details about user preferences for 12,294 anime titles, based on the input of 73,516 users. Each user has the ability to rate and add anime titles to their completed list, and this dataset comprises a collection of these ratings.

3.1.1. Number of Records

a. Anime Dataset



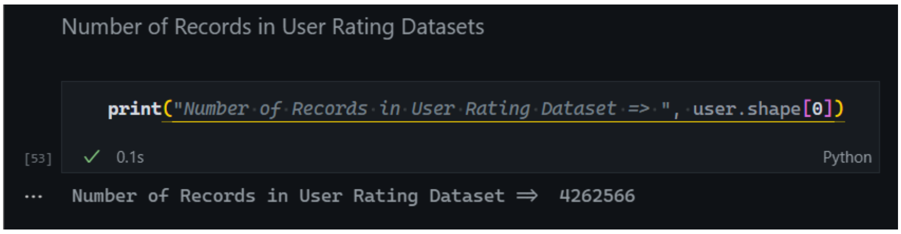
Number of Records in Anime Datasets

```
print("Number of Records in Anime Dataset=> ", anime.shape[0])
```

[52] ✓ 0.1s Python

```
Number of Records in Anime Dataset=>  12294
```

b. User Ratings Datasets



Number of Records in User Rating Datasets

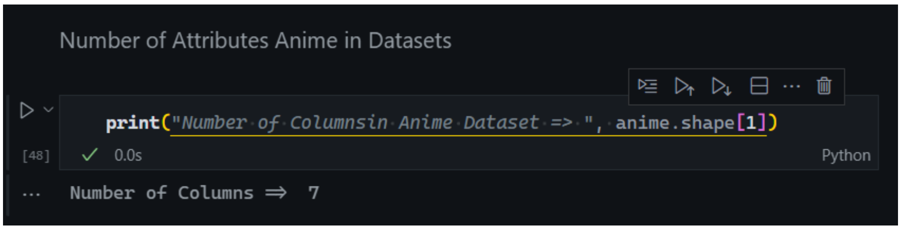
```
print("Number of Records in User Rating Dataset => ", user.shape[0])
```

[53] ✓ 0.1s Python

```
Number of Records in User Rating Dataset =>  4262566
```

3.1.2. Number of Attributes

a. Anime Dataset



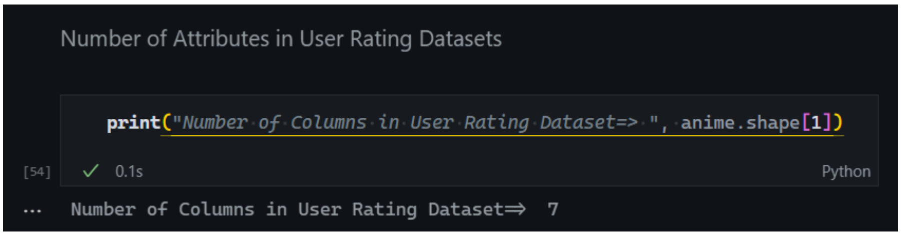
Number of Attributes Anime in Datasets

```
print("Number of Columnsin Anime Dataset => ", anime.shape[1])
```

[48] ✓ 0.0s Python

```
Number of Columns =>  7
```

b. User Ratings Dataset



Number of Attributes in User Rating Datasets

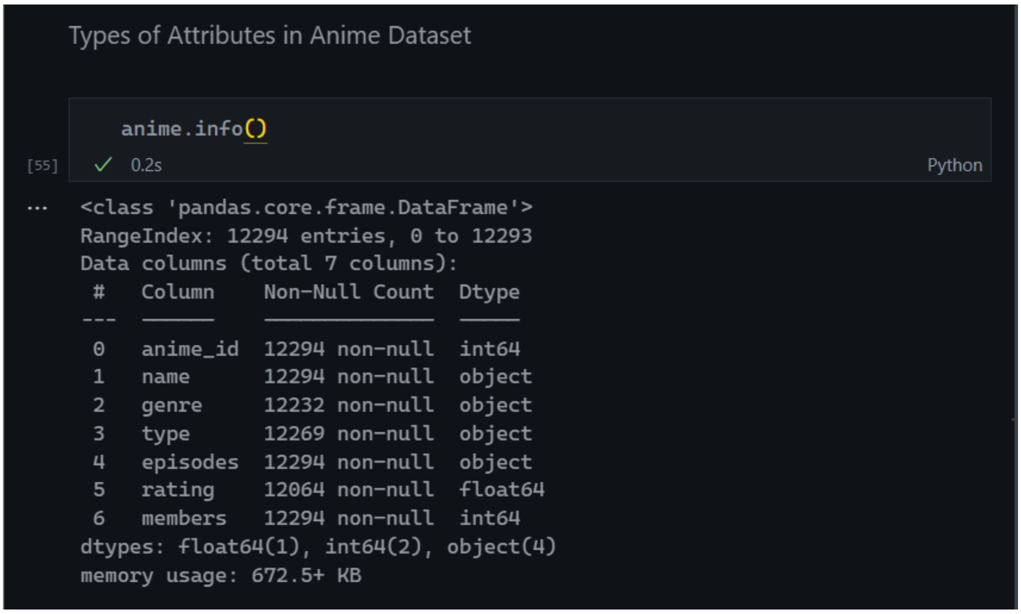
```
print("Number of Columns in User Rating Dataset=> ", anime.shape[1])
```

[54] ✓ 0.1s Python

```
Number of Columns in User Rating Dataset=>  7
```

3.1.3 Types of Attributes

a. Anime Dataset

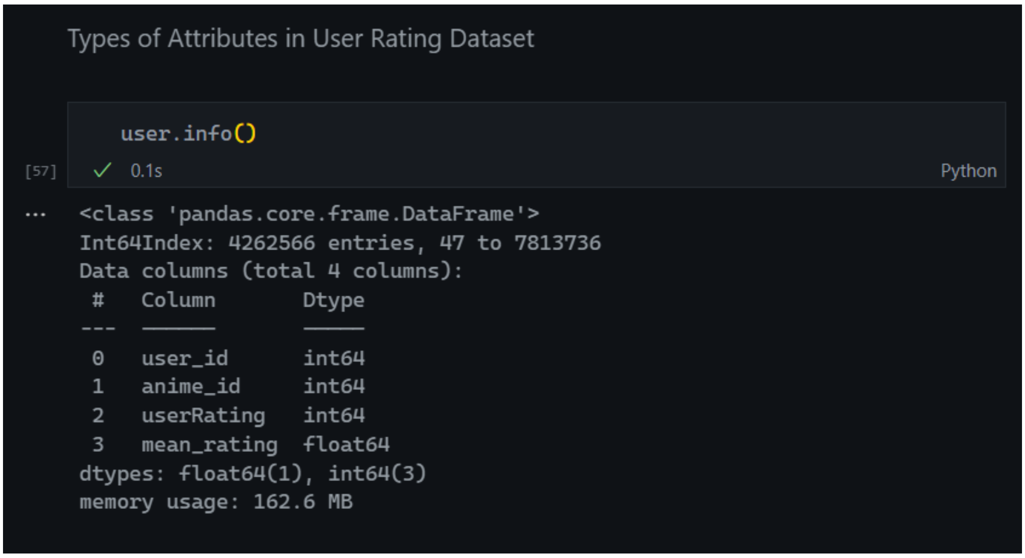


Types of Attributes in Anime Dataset

```python
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
```

b. User Ratings Datasets



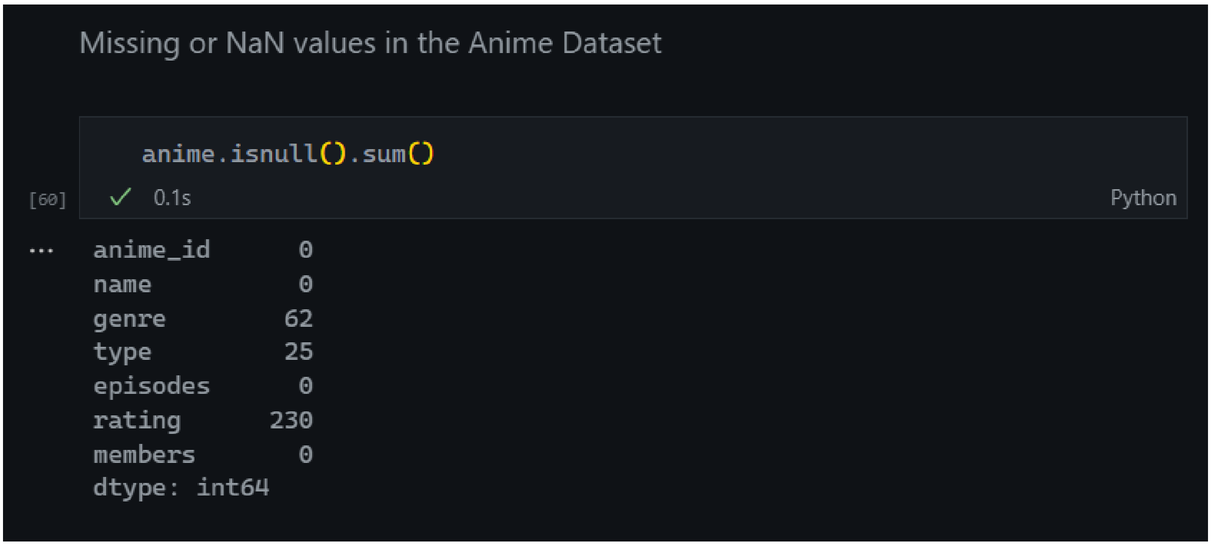
Types of Attributes in User Rating Dataset

```python
user.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4262566 entries, 47 to 7813736
Data columns (total 4 columns):
 #   Column       Dtype
---  ------       -----
 0   user_id      int64
 1   anime_id     int64
 2   userRating   int64
 3   mean_rating  float64
dtypes: float64(1), int64(3)
memory usage: 162.6 MB
```

3.1.4. Missing Values or Nulls

a. Anime Dataset

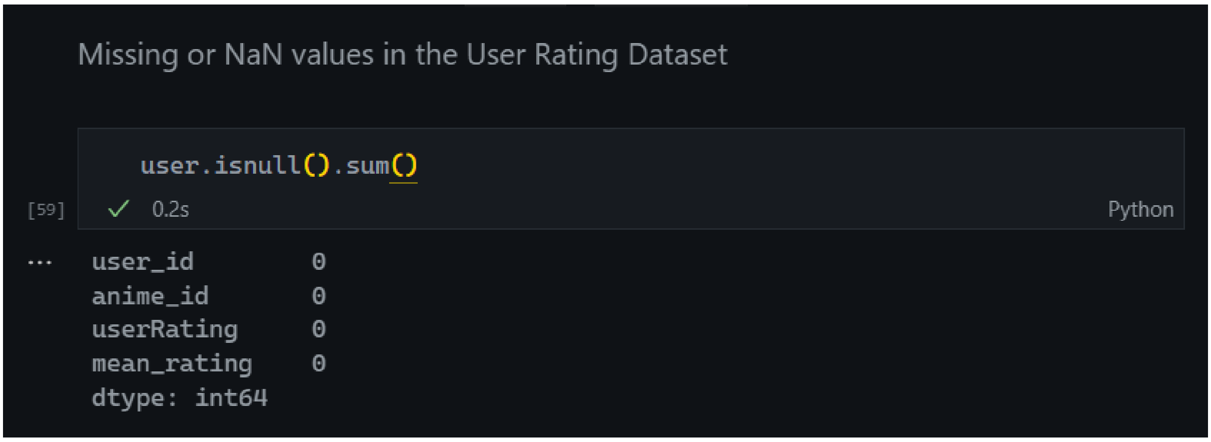


Missing or NaN values in the Anime Dataset

```
anime.isnull().sum()
```

```
anime_id      0
name          0
genre        62
type         25
episodes      0
rating      230
members       0
dtype: int64
```

b. **User Rating Dataset**



Missing or NaN values in the User Rating Dataset

```
user.isnull().sum()
```

```
user_id        0
anime_id       0
userRating     0
mean_rating    0
dtype: int64
```

3.1.5. Attributes Description

a. **Anime Dataset:** The dataset contains 7 attributes. The description of each is given below.

i. **anime_id** - myanimelist.net's unique id identifying an anime.
ii. **name** - full name of anime.
iii. **genre** - comma-separated list of genres for this anime.
iv. **type** - movie, TV, OVA, etc.
v. **episodes** - how many episodes in this show. (1 if movie).
vi. **rating** - an average rating out of 10 for this anime.
vii. **members** - number of community members that are in this anime's "Group".

b. **User Rating Dataset:** The dataset contains 7 attributes. The description of each is given below.

i. **user_id** - non-identifiable randomly generated user id.
ii. **anime_id** - the anime that this user has rated.
iii. **rating** - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

Chapter 4
DATA PREPROCESSING

Data preprocessing refers to the procedure of transforming raw data into a structured format that is appropriate for utilization in a data mining model. This encompasses a variety of steps, including cleaning the data to ensure that it is consistent and free of errors, as well as manipulating it to enhance model performance and accuracy.

The primary steps involved in data preprocessing include acquiring the dataset and importing the necessary libraries, extracting the independent and dependent variables from the dataset, addressing any missing data, encoding categorical data to numeric format, splitting the dataset into a training set and a testing set, and scaling the independent variables using a technique called feature scaling. This process ensures that the data is standardized and suitable for use in a machine learning model, leading to better accuracy and efficiency.

4.1. Handling Null Values

Missing data pertains to the absence of values or information for certain variables in a provided dataset. In Pandas, missing values are often designated as NaN. Addressing missing data in a dataset is crucial, since many machine learning models will generate errors if NaN values are utilized as inputs.

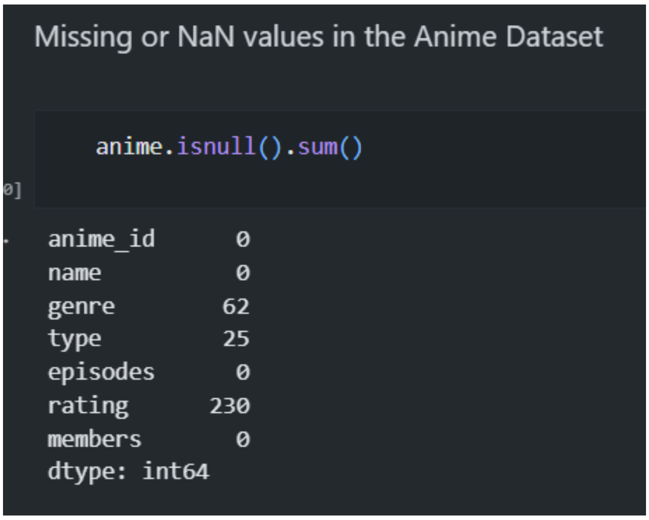
Therefore, it is imperative to fill in any gaps in data to ensure that machine learning models function accurately. By doing so, one can maximize the information available from the dataset and reduce the chance of errors in subsequent analyses.

There are multiple techniques available for addressing missing data in a dataset. Two of the most common methods are:

1. **Removing rows with missing data:** If a specific row contains missing values, the entire row, including all the features it contains, can be deleted from the dataset.

2. **Imputing missing values:** The missing data can be filled in with an appropriate value, such as the mean or median for numerical variables, or the mode for categorical variables. This approach can help preserve the integrity of the dataset and prevent the loss of valuable information.

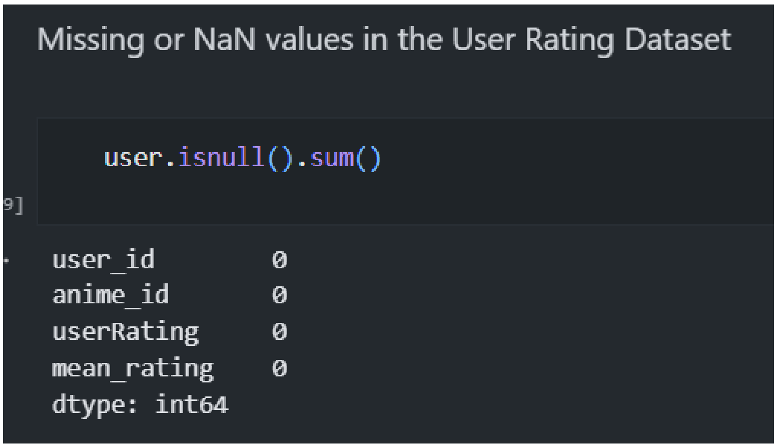
Checking the number of null values for each column.



Missing or NaN values in the Anime Dataset

```
anime.isnull().sum()
```

```
anime_id      0
name          0
genre        62
type         25
episodes      0
rating      230
members       0
dtype: int64
```



Missing or NaN values in the User Rating Dataset

```
user.isnull().sum()
```

```
user_id        0
anime_id       0
userRating     0
mean_rating    0
dtype: int64
```

Since, the count of null values is negligibly small compared to the size of dataset. Hence, null values are ignored in the dataset.

4.2 Feature Scaling

Feature scaling is a technique used in data preprocessing to standardize the independent variables of a dataset within a specific range. This process involves transforming the values of the variables so that they are on a similar scale and have comparable ranges.

The need for feature scaling arises because many machine learning algorithms perform poorly when the features have different scales. For instance, algorithms such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) rely on distance metrics between data points, and large differences in scale between the features can cause these algorithms to be biased towards the variables with larger scales.

Feature scaling can be achieved through different methods such as min-max scaling and standardization. In min-max scaling, the values of the variables are transformed to a range between 0 and 1. In standardization, the variables are transformed to have a mean of 0 and a standard deviation of 1. These techniques ensure that all the variables have a similar range, which can improve the accuracy and efficiency of the machine learning model.

4.2.1 Normalization

It is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1.

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, X = Original Value of feature

Xmin = Minimum value of column

Xmax = Maximum value of Column

4.2.2 Standardization

It is a scaling technique where the values have a unit standard deviation and are centred around the mean. Since the attribute's mean is now 0, the distribution that results has a unit standard deviation.

$$X^{'} = \frac{X - \mu}{\sigma}$$

Here, μ is the mean of the feature values and σ is the standard deviation of the feature values.

4.3 Feature Selection And Conversion

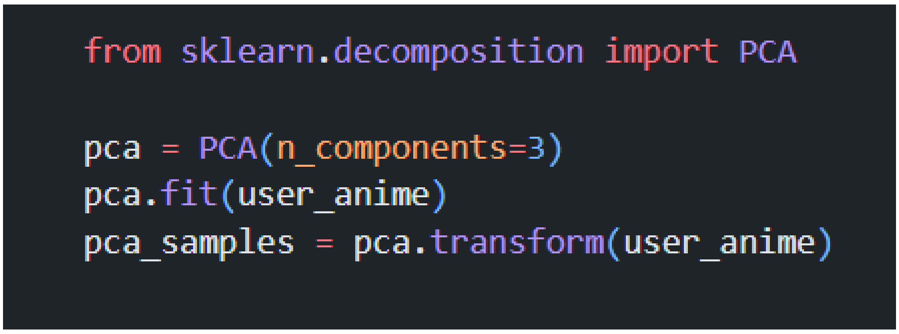
During feature selection, the objective is to identify and select the features that have the most significant impact on the outcome variable or the predicted result. When the dataset contains irrelevant or redundant features, it can adversely affect the accuracy of the model. Therefore, the process of feature selection is crucial for achieving high accuracy in predictive modeling.

As an example of a feature selection method utilised in this project, consider what follows:

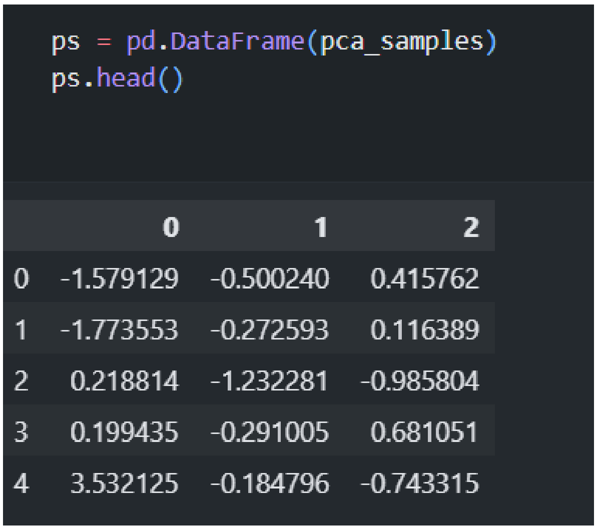
Principal Component Analysis

Our initial set of variables is transformed into a new set of variables through principal component analysis, which creates a linear combination of the first set of variables.

Data dimension reduction is the key objective in order to cluster and visualise the data.

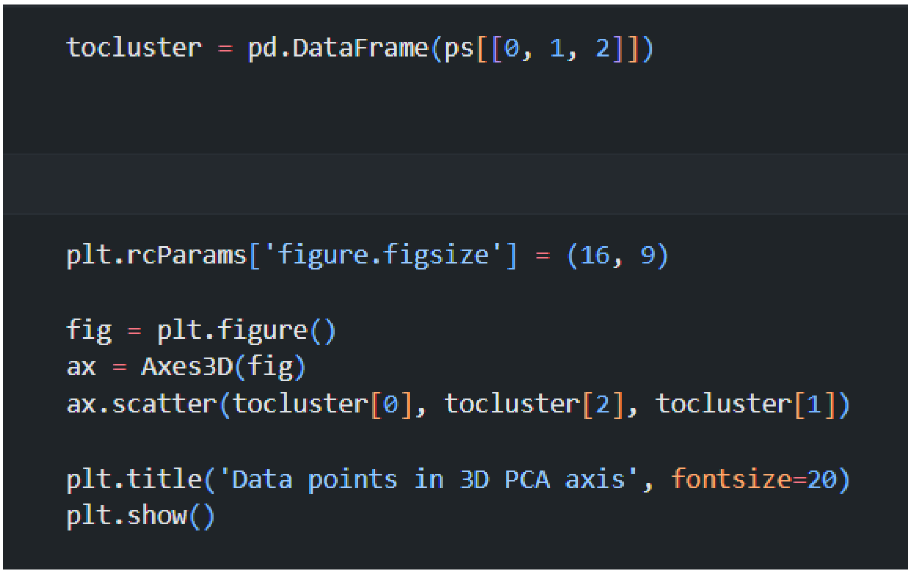


```python
from sklearn.decomposition import PCA

pca = PCA(n_components=3)
pca.fit(user_anime)
pca_samples = pca.transform(user_anime)
```



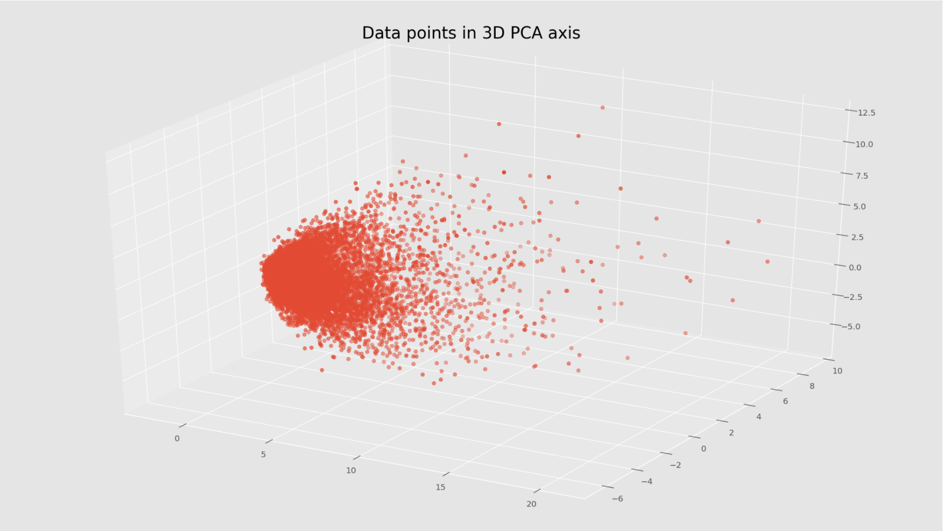
```python
ps = pd.DataFrame(pca_samples)
ps.head()
```

	0	1	2
0	-1.579129	-0.500240	0.415762
1	-1.773553	-0.272593	0.116389
2	0.218814	-1.232281	-0.985804
3	0.199435	-0.291005	0.681051
4	3.532125	-0.184796	-0.743315

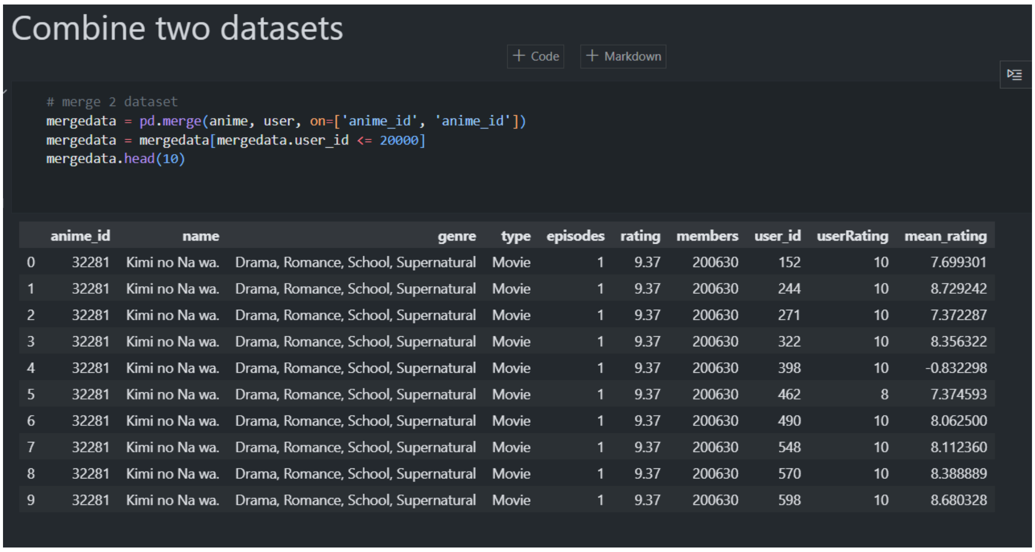


```python
tocluster = pd.DataFrame(ps[[0, 1, 2]])
```

```python
plt.rcParams['figure.figsize'] = (16, 9)

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(tocluster[0], tocluster[2], tocluster[1])

plt.title('Data points in 3D PCA axis', fontsize=20)
plt.show()
```

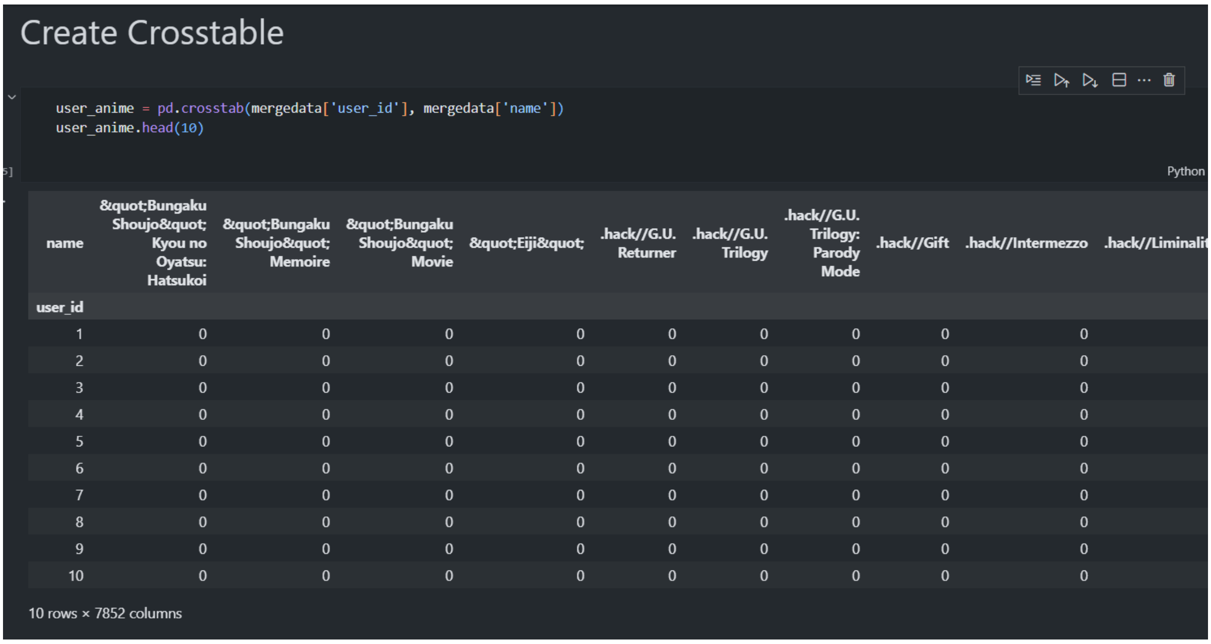


Conversion


Combine two datasets

```
# merge 2 dataset
mergedata = pd.merge(anime, user, on=['anime_id', 'anime_id'])
mergedata = mergedata[mergedata.user_id <= 20000]
mergedata.head(10)
```

	anime_id	name	genre	type	episodes	rating	members	user_id	userRating	mean_rating
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	152	10	7.699301
1	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	244	10	8.729242
2	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	271	10	7.372287
3	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	322	10	8.356322
4	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	398	10	-0.832298
5	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	462	8	7.374593
6	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	490	10	8.062500
7	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	548	10	8.112360
8	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	570	10	8.388889
9	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	598	10	8.680328



Create Crosstable

```python
user_anime = pd.crosstab(mergedata['user_id'], mergedata['name'])
user_anime.head(10)
```

name	&quot;Bungaku Shoujo&quot; Kyou no Oyatsu: Hatsukoi	&quot;Bungaku Shoujo&quot; Memoire	&quot;Bungaku Shoujo&quot; Movie	&quot;Eiji&quot;	.hack//G.U. Returner	.hack//G.U. Trilogy	.hack//G.U. Trilogy: Parody Mode	.hack//Gift	.hack//Intermezzo	.hack//Liminalit
user_id										
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	

10 rows × 7852 columns

4.4 Data Sampling And Subsetting

Sampling is the process of examining a small subset of all the data to find the important information in the bigger set.

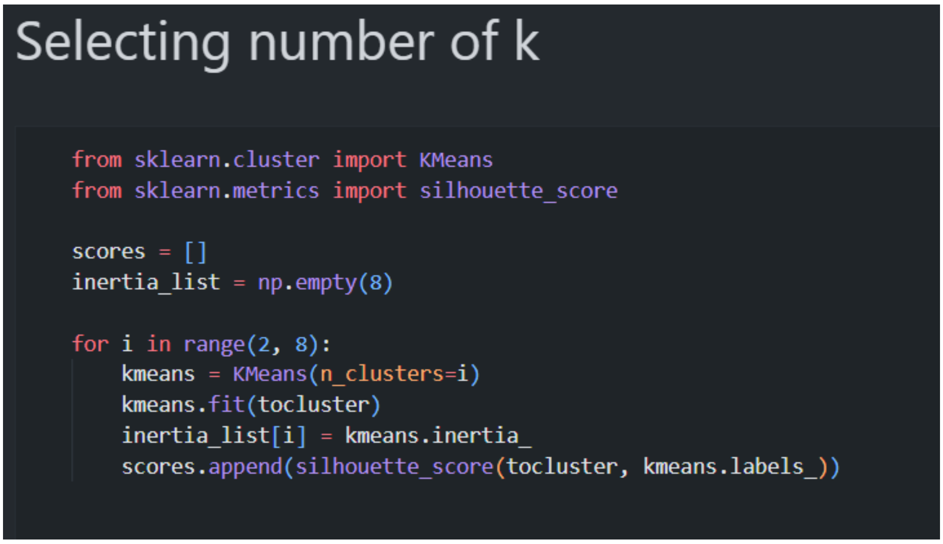
Data splitting is the division of a dataset into two or more subsets. The data is separated into two parts; the first part is used to train the model, and the second part is used to assess or test the data.

It is possible to divide the data into training and testing sets in a variety of methods. Some methods of data splitting include random subsampling, the Hold-out method, and cross-validation.

Chapter 5
Building Models

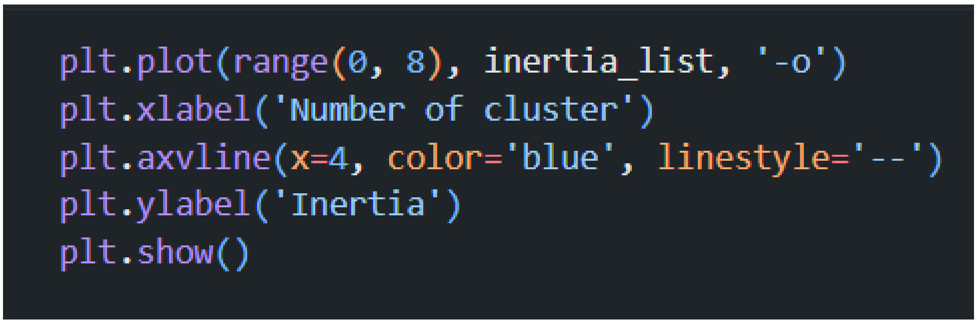
In this research, the classification models K-Means are employed.

5.1 Model 1: K-means

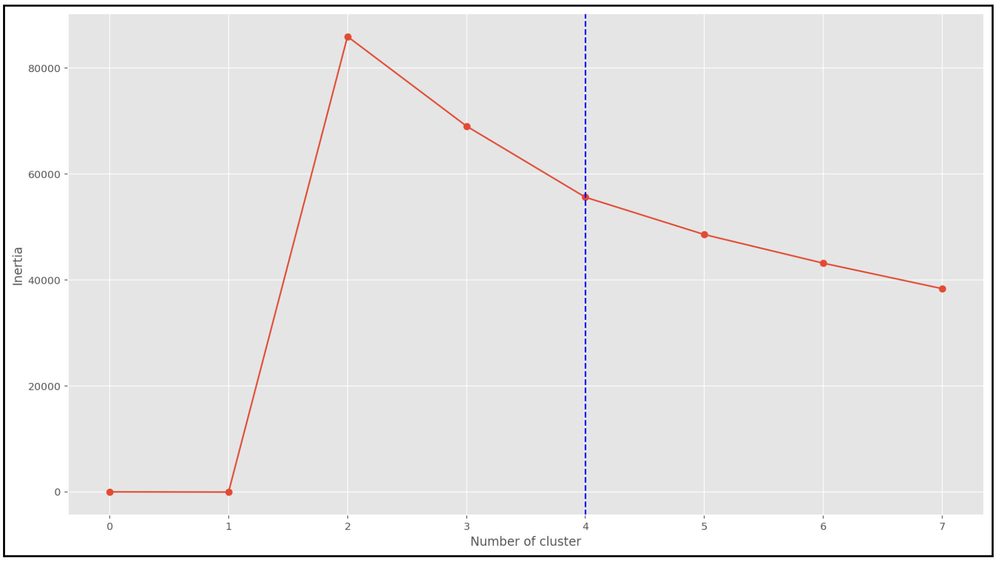


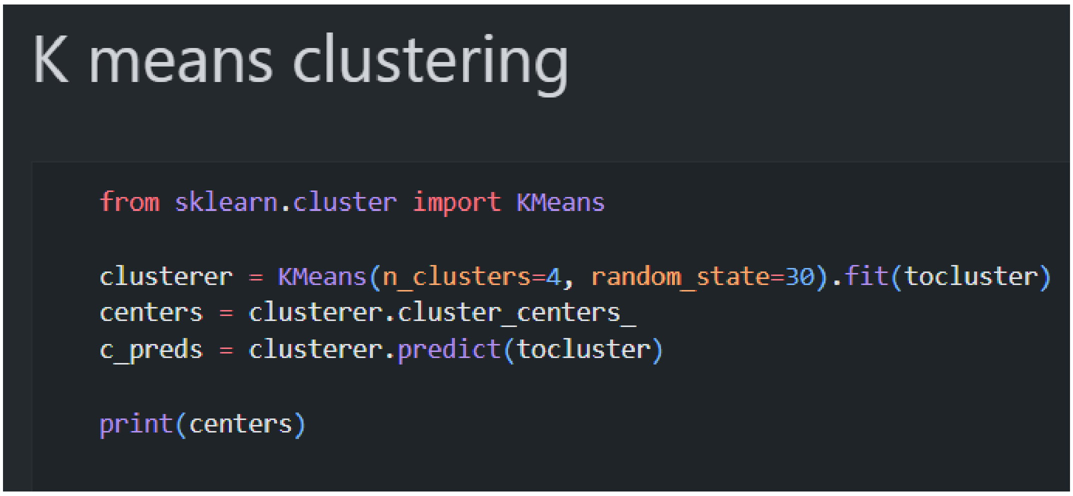
Selecting number of k

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

scores = []
inertia_list = np.empty(8)

for i in range(2, 8):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(tocluster)
    inertia_list[i] = kmeans.inertia_
    scores.append(silhouette_score(tocluster, kmeans.labels_))
```



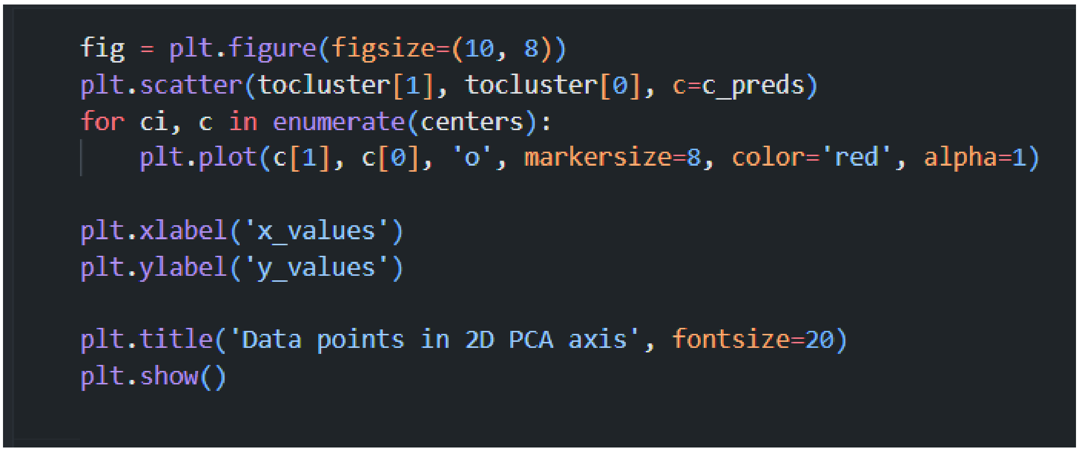
```
plt.plot(range(0, 8), inertia_list, '-o')
plt.xlabel('Number of cluster')
plt.axvline(x=4, color='blue', linestyle='--')
plt.ylabel('Inertia')
plt.show()
```



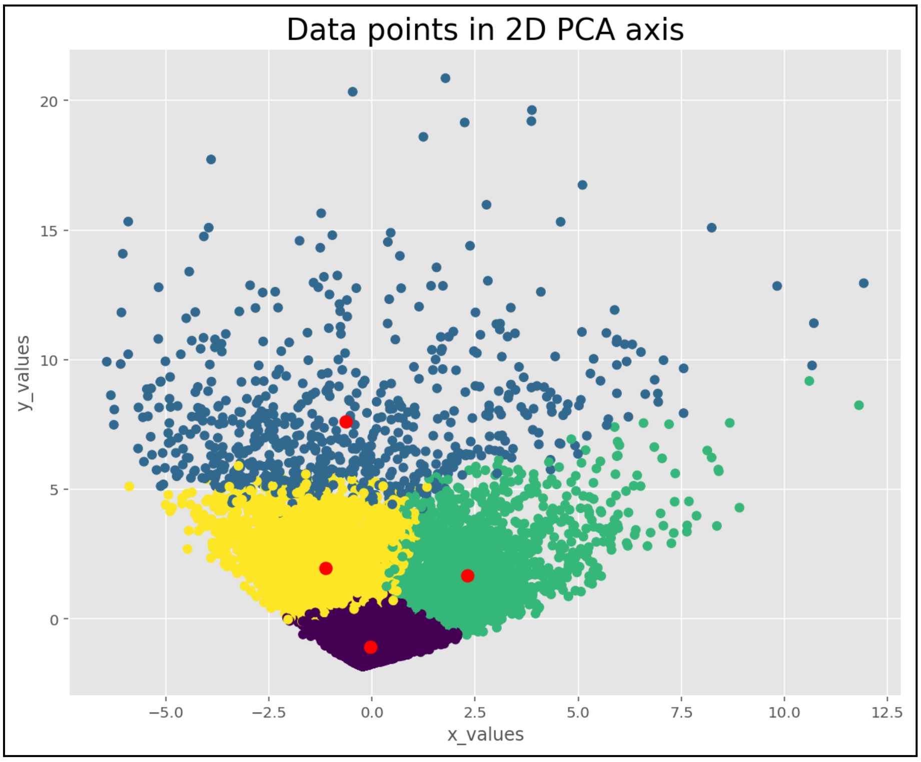


K means clustering

```
from sklearn.cluster import KMeans

clusterer = KMeans(n_clusters=4, random_state=30).fit(tocluster)
centers = clusterer.cluster_centers_
c_preds = clusterer.predict(tocluster)

print(centers)
```



```
fig = plt.figure(figsize=(10, 8))
plt.scatter(tocluster[1], tocluster[0], c=c_preds)
for ci, c in enumerate(centers):
    plt.plot(c[1], c[0], 'o', markersize=8, color='red', alpha=1)

plt.xlabel('x_values')
plt.ylabel('y_values')

plt.title('Data points in 2D PCA axis', fontsize=20)
plt.show()
```


Data points in 2D PCA axis
y_values
x_values
20
15
10
5
0
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5

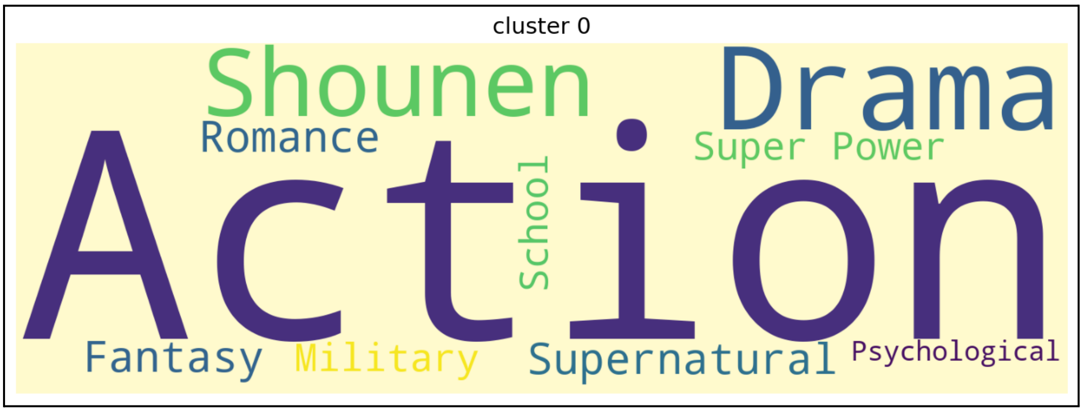
Chapter 6
Model Evaluation And Results

6.1 METRICS

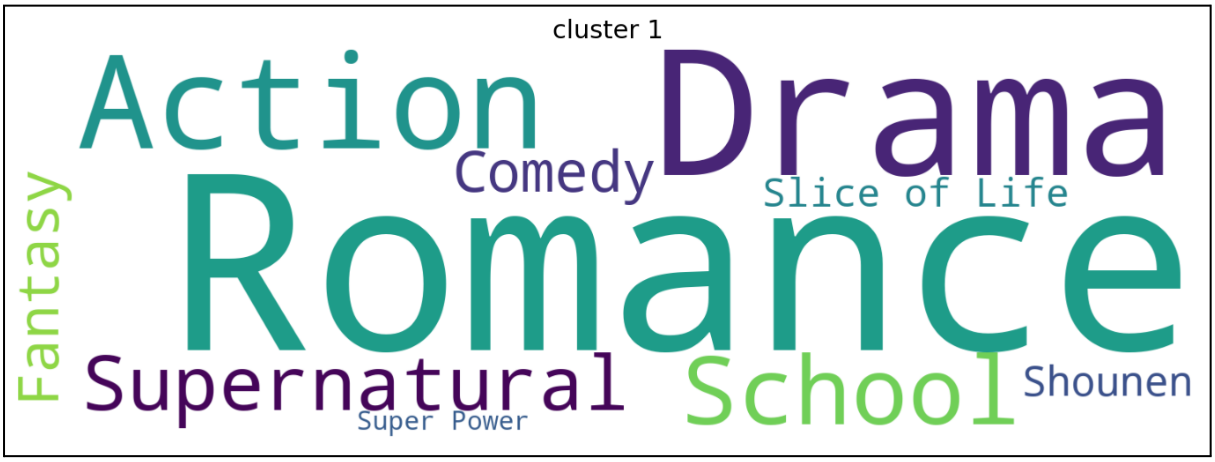
A variety of evaluation measures can be used to gauge the effectiveness of a machine learning model. Using hypothetical data, it attempts to gauge a model's generalisation accuracy.

6.2. EXPERIMENTAL RESULTS AND COMPARISON

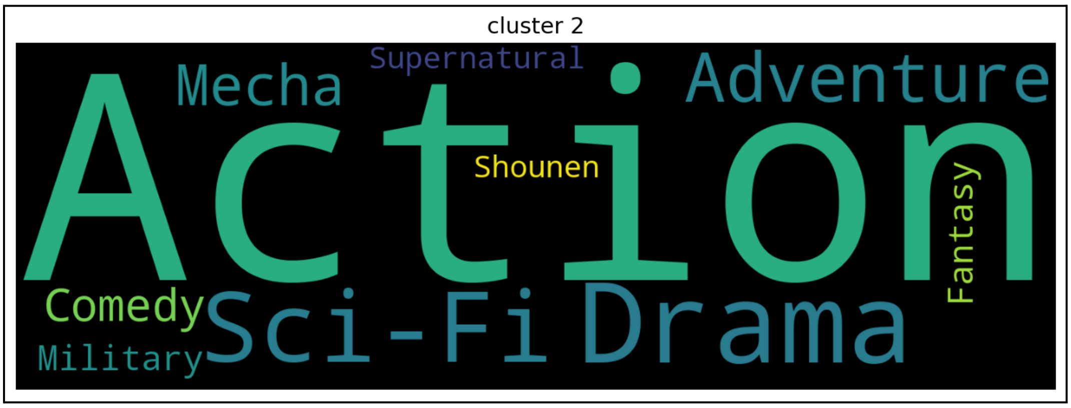
- Favourite genre of cluster 0



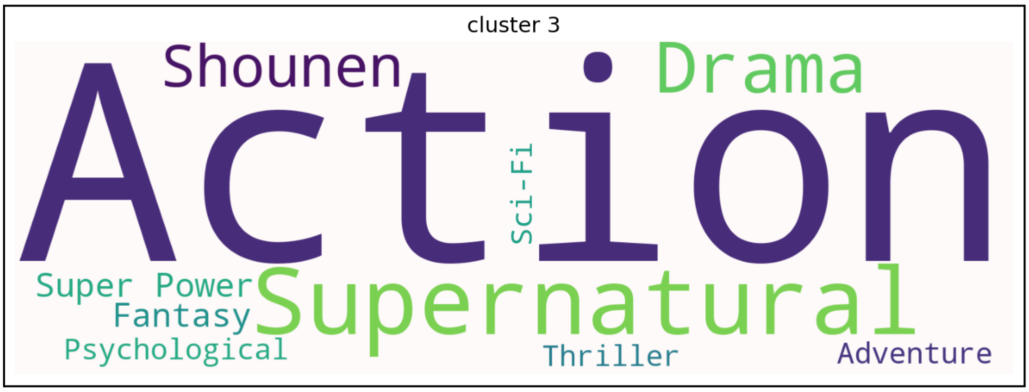
- Favourite genre of cluster 1



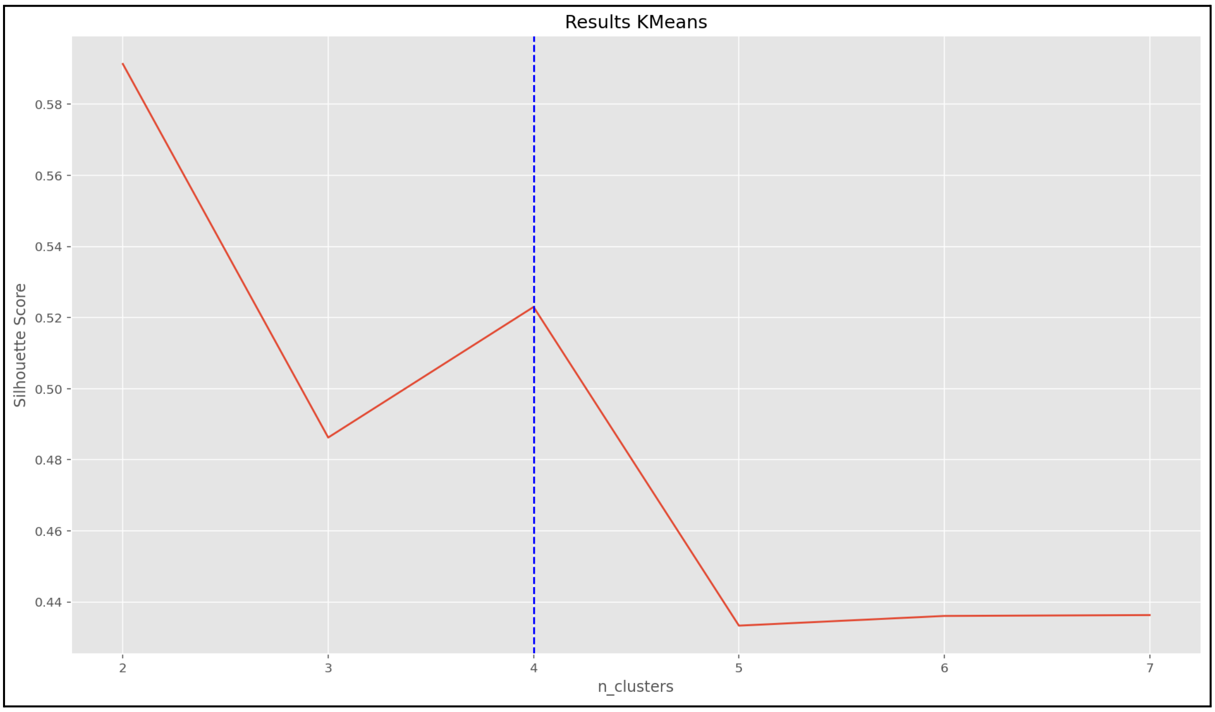
- Favourite genre of cluster 2



- Favourite genre of cluster 3



Silhouette Score:



Chapter 7
Inferences And Conclusion

1. The model used in this project is K mean.
2. Four clusters are formed on the basis of user ratings.
3. The Silhouette Score came out to be approx 0.52 when 4 clusters are formed.
4. This project concludes that according to the dataset used while clustering 4 clusters will be formed with the metrics called silhouette score = 0.52.

Reference

1. Source Code: https://github.com/khatrijiraj/data-mining-anime-recommendation

2. https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database

3. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education.

4. Concepts and Techniques, 3nd edition,Jiawei Han and Micheline Kamber

5. A Tutorial Based Primer, Richard Roiger, Michael Geatz, Pearson Education 2003.

6. Introduction to Data Mining with Case Studies, G.K. Gupta, PHI 2006

7. Insight into Data mining: Theory and Practice, Soman K. P., DiwakarShyam, Ajay V., PHI 2006