# **Introduction**

## Motives

While searching for an interesting topic for our final project, we are looking for a meaningful project that can be fun to work on. We discussed our hobbies among our group members and concluded that we all like watching anime. As anime watchers, we all tend to struggle to find a good show. It takes countless hours to find the right show to get you hooked up.

## Project Introduction

We aim to create a solution that allows you to search anime effortlessly and filter down the results based on similar animes you have previously watched.

# **Background Research**

While searching for appropriate ways to filter down and find matches that are similar to user input, we found a few methods to filter down results, and the one that caught our interest was [clustering](https://arxiv.org/abs/2109.12839).

We also took a look at other methods such as [Collaborative Filtering](https://www.researchgate.net/publication/200121027_Collaborative_Filtering_Recommender_Systems), [Collaborative Filtering in anime recommendation](https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012057/meta) systems, [Content-Based Filtering](https://www.researchgate.net/publication/236895069_Content-Based_Recommendation_Systems), and [Hybrid filtering](https://www.researchgate.net/publication/321281871_Hybrid_Recommender_Systems_A_Systematic_Literature_Review).  
We concluded that our ideal solution would involve clustering with collaborative filtering to improve accuracy. In addition, we found more research paper that supports [Collaborative Filtering for anime recommendation](https://www.researchgate.net/publication/342690182_Collaborative_Recommendation_System_in_Users_of_Anime_Films). Lastly, we found some research papers that are outside our course's scope and would like to use in the future, such as [extracting features from full text](https://ieeexplore.ieee.org/document/9315363) to support more rich recommendations.

In addition, we took inspiration from a few existing anime recommendation sites such as [AniBrain.ai](https://anibrain.ai/) and [RandomAnime.org](http://randomanime.org/). We took time to brainstorm how these current sites work.

**Analysis & Result**

## Introduction

Our project aims to offer an anime list to a new user. Thus we went with the collaborative filtering recommendation system, which suggests items based on the user's shared preferences. As a result, the query is transformed into a clustering question. Finally, we choose the unsupervised model K-Means as a clustering strategy for the modeling process.

## K-Means

### Brief Introduction

A cluster is a group of data points that have been grouped due to some commonalities. You'll specify a goal number, k, for the number of centroids in the dataset. The cluster's center is represented by a centroid, which can be either imaginary or real. By lowering the in-cluster sum of squares, each data point is assigned to each of the clusters.

To put it another way, the K-means algorithm finds k centroids and then assigns each data point to the closest cluster while keeping the centroids as small as feasible. The [K-means](https://ieeexplore.ieee.org/document/5453745) technique in data mining starts with the initial set of randomly picked centroids, which serve as the starting points for each cluster. It then performs iterative (repetitive) computations to optimize the centroids' placements.

It halts creating and optimizing clusters when either:

* The centroids have stabilized — there is no change in their values because the clustering has been successful.
* The defined number of iterations has been achieved.

### Data Description

Type of recommendation System

1. Collaborative Filtering: the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. Applications of collaborative filtering typically involve extensive data sets.
2. Content-Based Filtering: This is a recommender system that attempts to guess what a user may like based on that user's activity. Content-based filtering recommends using keywords and attributes assigned to objects in a database (e.g., items in an online marketplace) and matching them to a user profile.
3. A hybrid recommendation system is a special recommendation system that can be considered a combination of the content and collaborative filtering method.

Data preprocessing

* Data Cleaning: We begin recognizing and eliminating records from our database that are corrupt or erroneous. We will just choose the most important ones in this situation. We can drop a few missing data points if there are a few, and we can average if there are a lot. The data set should be consistent with other similar data sets in the system after cleansing. User entry errors, transmission or storage damage, or differing data dictionary definitions of equivalent things in other stores may have caused the discrepancies discovered or eliminated.
* Data Transformation Our data is being converted from one format to another in an inefficient manner. Transforming raw data into a clean and useful form and converting CV are two phases in data transformations. An analyst will determine the structure of the data, do data mapping, extract the data from the source, execute the transformation, and store the data in the appropriate database during data transformation.
* Data Reduction: This features a capacity optimization strategy that reduces data to its most basic form in order to free up storage space on a storage device. There are various techniques to minimize data, but the concept is simple: fit as much data as possible into physical storage to maximize capacity.

EDA

In this project, we'll use two data sets: anime.csv, which contains information about the anime, and data.csv, which contains information about the data. The user's rating of the anime is included in the second. During the EDA process, we will look for hypotheses (fig 1) between each feature from a separate dataset. A box plot (also known as a box-and-whisker plot) depicts the distribution of quantitative data in a way that makes it easier to compare variables. The box depicts the dataset's quartiles, while the whiskers extend to depict the remainder of the distribution.

### 

### Fig.1

### Parameters in the Scikit-learn API

Source: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

1. N\_clusters: int, default=8

The number of clusters to form as well as the number of centroids to generate.

1. Init, default=’k-means++’

‘K-means++’: selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

‘random’: choose n\_clusters observations (rows) at random from data for the initial centroids.

1. n\_initint, default=10

The number of times the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n\_init consecutive runs in terms of inertia.

1. Max\_iter: int, default=300

The maximum number of iterations of the k-means algorithm for a single run.

1. Tol: float, default=1e-4

Relative tolerance with regards to the Frobenius norm of the difference in the cluster centers of two consecutive iterations to declare convergence.

1. Verbose :int, default=0

Verbosity mode.

1. Random\_state: int, RandomState instance or None, default=None

Determines random number generation for centroid initialization. Use an int to make the randomness deterministic.

1. Copy\_x: bool, default=True

When pre-computing distances it is more numerically accurate to center the data first. If copy\_x is True (default), then the original data is not modified. If False, the original data is modified, and put back before the function returns, but small numerical differences may be introduced by subtracting and then adding the data mean. Note that if the original data is not C-contiguous, a copy will be made even if copy\_x is False. If the original data is sparse, but not in CSR format, a copy will be made even if copy\_x is False.

1. Algorithm: {“lloyd”, “elkan”}, default=”lloyd”

K-means algorithm to use. The classical EM-style algorithm is "lloyd". The "elkan" variation can be more efficient on some datasets with well-defined clusters, by using the triangle inequality. However, it’s more memory intensive due to the allocation of an extra array of shapes (n\_samples, n\_clusters).

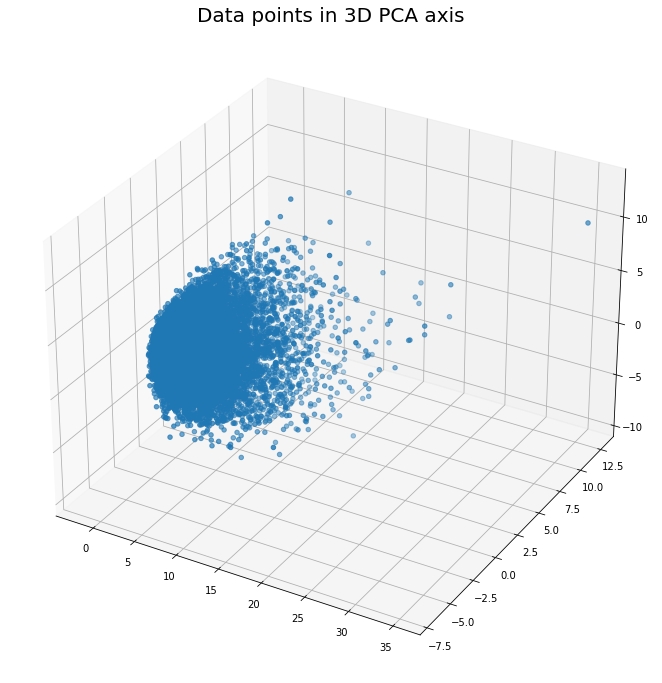
In our project, we use the parameters: n\_clusters, Init, Random\_state, and Algorithm. However, after we run the model, the parameters may change the result is the number of clusters, others are already perfect with the default value.

clusterer = KMeans(n\_clusters=4,random\_state=30).fit(tocluster)

## PCA (Principal component analysis)

PCA is used to reduce the data's dimensionality and project it into a lower-dimensional environment. Before applying the SVD, the input data is centered but not scaled for each feature.

Basically, we utilize this for dimensional reduction and modeling because if we apply PCA to our data, it will be easier to visualize. The data visualization in 3D is shown in the image below.



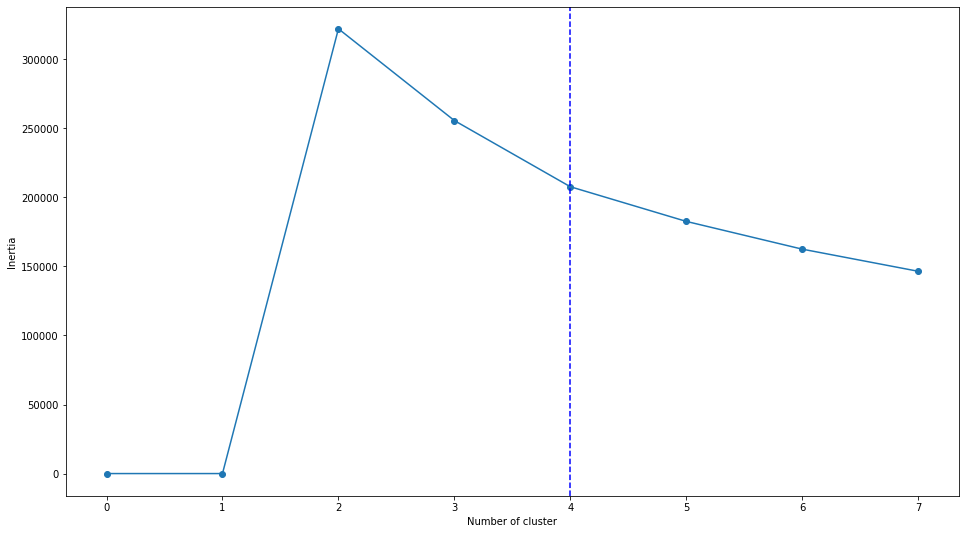
## Find the best K value

We must determine the K values for the number of centroids in order to create an effective clustering model. The basic idea is to calculate the distance between each centroid and each data point in the group to the centroid. There are several ways to make a decision, but they all have the same idea to evaluate the model: the basic idea is to calculate the distance between each centroid and each data point in the group to the centroid. The approach we used to find the number of K was to utilize the elbow.

### Elbow Method

In k-means clustering, the elbow method is used to figure out how many clusters are needed. The elbow method plots the cost function value resulting from various k values. As you might expect, as k grows, average distortion decreases, each cluster has fewer constituent instances, and the instances are closer to their respective centroids. As k grows larger, however, the average distortion improves less. The elbow is the value of k at which the improvement in distortion diminishes the most and at which we should stop dividing the data into more clusters.

* Distortion: It is calculated as the average of the squared distances from the cluster centers of the respective clusters. Typically, the Euclidean distance metric is used.
* Inertia: It is the sum of the squared distances of samples to their closest cluster center.



The line chart for our project shows the inertia value vs. the number of clusters. We determined from the graph that when the number of clusters reaches four, the graph has a non-obvious turning point, hence the k value should be three or four. We'll utilize a variety of factors to get the optimum k value because we can't define it.

## Clustering Validation

For supervised learning, we can use accuracy and cross-validation to verify the generalization ability of the model, but for the verification process of the clustering model, we need to apply a different verification process.

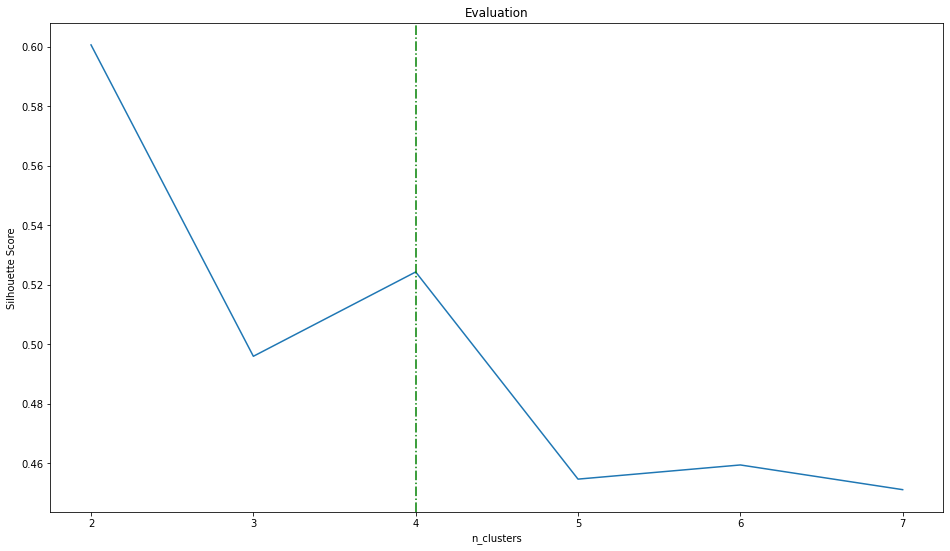
For this question, we decided to use these two criteria to evaluate the model:

1. Silhouette Coefficient
2. Calinski-Harabasz Index

### Silhouette Coefficient

Each samples mean intra-cluster distance (a) and mean nearest-cluster distance (b) are used to determine the Silhouette Coefficient. (b - a) / max is the Silhouette Coefficient for a sample (a, b). To be clear, b is the distance between a sample and the closest cluster to which it does not belong.

1 is the best value, while -1 is the lowest. Overlapping clusters are indicated by values close to zero. Negative numbers usually imply that a sample was allocated to the incorrect cluster because another cluster is more comparable.

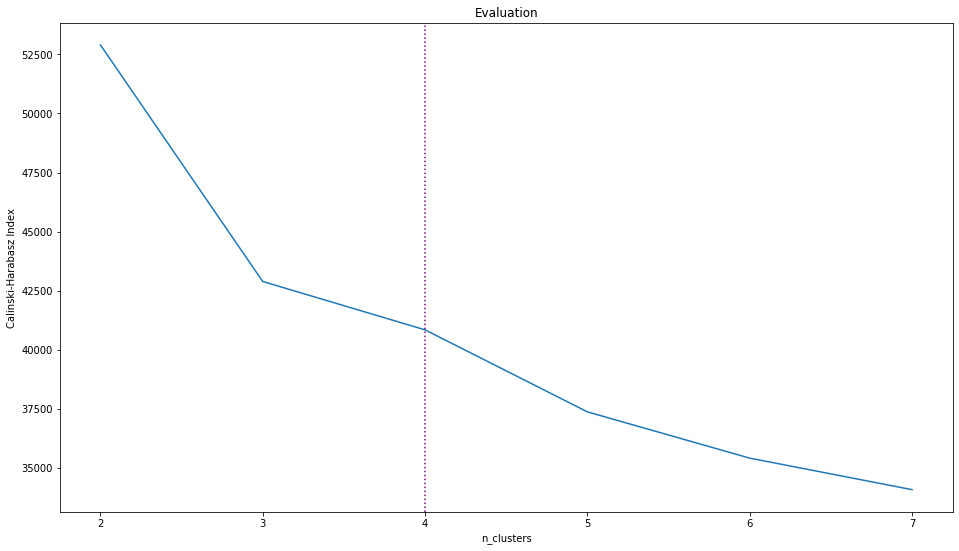


We want to use Silhouette Score to both analyze the model and discover the optimum k value in this procedure. Our project's Silhouette Score vs. K clusters is depicted in the graph above. As you can see from the graph, there is a sharp turning point where the location K equals four. After then, as the number of clusters reaches infinite, the Silhouette score approaches zero.

### Calinski-Harabasz Index

The Variance Ratio Criterion is another name for it.

The score is calculated by dividing the total of between-cluster and within-cluster dispersion by the number of clusters. And the greater the numbers, the better.

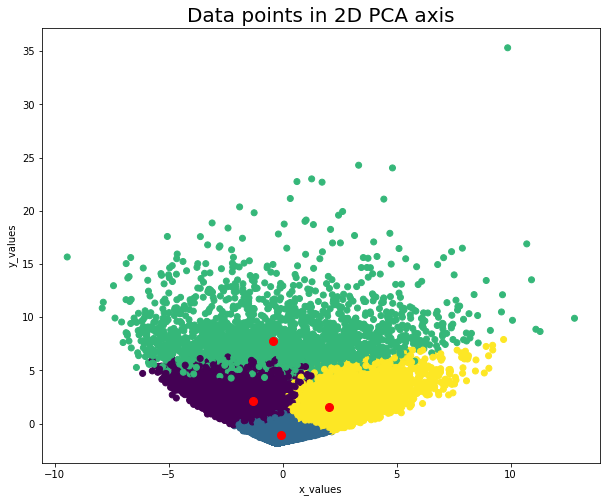
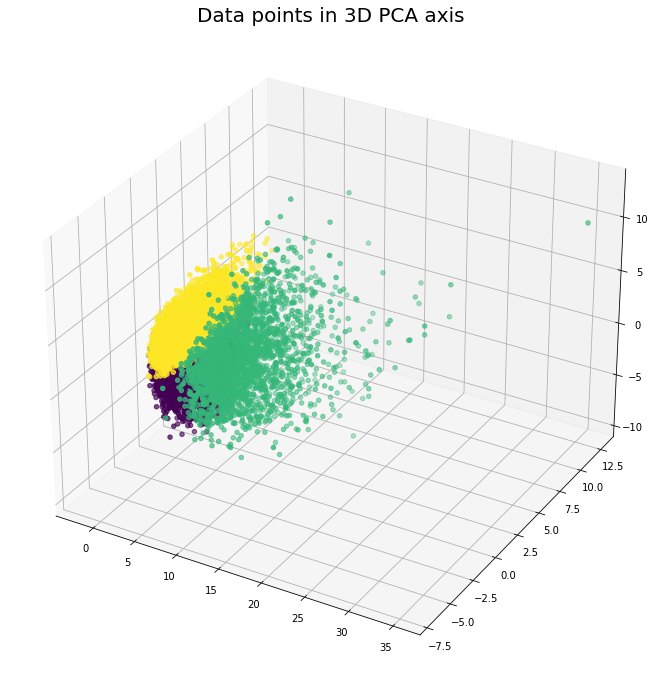


The Calinski-Harabasz Index is plotted against the standard deviation. According to our study, the main flow is comparable to the Silhouette Coefficient graph, but the index when clusters equal 4 has a non-obvious concave down shape, indicating a turning point.

Finally, we evaluate three criteria to determine the optimum k value for the models, and we find that 4 is the best number for the data points we have.

## Visualization

### Data Points



The two charts above show the data points after clustering. There are four clusters, and four red dots mark each centroid. As you can see from the charts, the data points are centered on the centroid. However, the border of each cluster is not clearly distinguished, so we believe that this is because the points in the clusters are highly intensive in 3D or 2D. And for each cluster, we made a word cloud chart to illustrate the genre inside of the anime list. As you can see from the graph.

The first time we visualize the cluster is like this:









As you can see, the genres overlap. That makes sense since most of the animes have action and comedy animes as we defined them before in the EDA process. However, in order to improve this problem, we hereby delete the action and comedy genres from the anime. Then we have a better result.

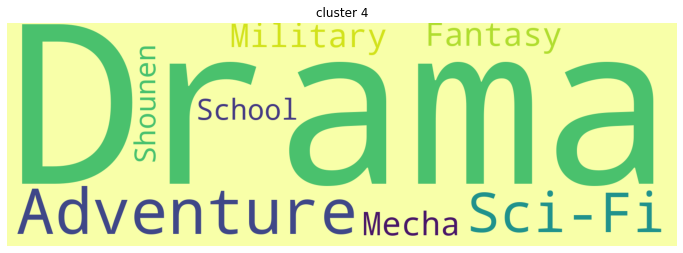
For each cluster:

1. Cluster 1: People like the anime with genres Fantasy, Shounen, and drama,
2. Cluster 2: People like the anime with the genre Drama with a small portion of others like Romance and School
3. Cluster 3: Users in this group prefer the animes with Supernatural, Drama, and Romance.
4. Cluster 4: Users prefer the animes with Drama, Adventure, and Sci-Fi









# **Conclusion and Future Plans**

In conclusion, we found that clustering can be used to generate accurate results based on user input. However, one of the major problems we still face is narrowing down the results into the most appropriate theme. Our current data doesn’t have any attributes to use for this problem.

In the future, we would like to implement more ways to filter down and narrow the results, add more features and hopefully create web UI.

In addition, we would like to gather feedback from users and improve the results based on user accuracy feedback. After gathering enough results, we could build our own dataset that can be used. One of the things we are also looking to explore is adding themes to each show manually or by using the community this way, we can add another important feature that we are missing in our dataset.