# Research

The purpose of the Research document is to examine what technologies and documentation are available in assisting with the creation of a web-application music recommender system. Within this research document there will be an array of information on what recommender systems are, the different types of recommender systems and the algorithms that are contained within these systems.

Over the last decade, recommender systems have become instrumental in one’s day to day life. With the rise of YouTube, Netflix, Amazon and many other web services, recommender systems are unavoidable from e-commerce to online advertisement. In short, recommender systems are algorithms aimed at suggesting relevant items to users e.g what movies to watch, books to read, music to listen to etc. Recommender systems are critical in most industries as they possess the ability to generate a substantial amount of income when they perform efficiently.

## Collaborative vs Content-Based Filtering

### Collaborative filtering methods

Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items to produce new recommendations. These interactions are stored in the so-called “user-item interactions matrix”.

Text

Description automatically generated

Figure 4: user-item interactions matrix.

The main aim of these collaborative based methods is that these past user-item interactions are sufficient in generating new recommendations for similar users/items and make predictions based on these proximities. Collaborative filtering is divided into sub-categories known as memory and model-based approaches. Memory based approaches works directly with the knowledge of previous interactions and rely on the ‘Nearest Neighbour’ search. An example of this would be to find the closest users from a user of interest and suggest the most popular items among these neighbours. Model-based approaches assume an underlying generated model that can explain the various user-item interactions and try to generate new recommendations based on the model itself.

A picture containing text

Description automatically generated

Figure 5: Collaborative filtering methods

The key advantage of collaborative filtering approaches is that they require no prior information about users or items, and this allows them to be applied in multiple scenarios. Furthermore, the more users that interact with items on a website, the more recommendations become accurate. Within a fixed set of users and items, new interactions that are recorded bring new information over time and make the recommender system more effective. However, as collaborative filtering only considers past interactions to make recommendations it suffers from the ‘cold start’ problem. “It is impossible to recommend anything to new users or to recommend a new item to any users and many users or items have too few interactions to be efficiently handled”. This issue can be resolved with a number of temporary solutions whilst the website is in its infancy: Recommending random items to new users/ new items to random users, recommending popular items or new items to most active users (known as the ‘maximum expectation strategy’) or even using non-collaborative methods for the early life of the user.

### Memory-based collaborative filtering

As discussed above; memory-based is the standard in collaborative and commonly features neighbourhood based algorithms. These algorithms produce similarity scores between two items/users which then generate an array of predictions based on these. The main cited disadvantage of this type of filtering is its performance with large datasets. Another issue is that this area of filtering can be sensitive to sparse data – data with a large number of null cells. Datasets that include books or music can be considered sparse data as users may not have rated every index in the set or songs from the 1920’s-1950’s may have missing information.

Chart, scatter chart

Description automatically generated

Figure 6: Visual demonstrating how K-Nearest neighbour uses Euclidean distance to locate similar items in vector space.

### Model-based collaborative filtering

Model based collaborative rely on user-item interactions and a latent model to explain these interactions. Matrix factorisation algorithms consists in decomposing the large user-items interaction matrix into a product of two small matrices: a user-factor matrix (contains users’ representations) that multiplies a factor-item matrix (contains items representations).

Diagram

Description automatically generated

Figure 7: Matrix factorisation method

### Content based methods

Unlike collaborative approaches that require user-item interactions, content-based approaches use additional information about users/items. If we take a music recommender system as an example, this additional information can be the artist, the music genre, the duration of the song etc. The main goal of content-based methods is to attempt to construct a model that are based around the available ‘features’, that explain the observed user-item interactions.

Diagram

Description automatically generated

Figure 8: Content based methods

One major advantage that content-based methods have over collaborative is that is suffers far less from the cold start problem. New users/items can be described by their characteristics (content) and so accurate suggestions can be done for these new entities. The main type of content-based filtering happens in the form of ‘Cosine Similarity’ – Which measures the similarity of two vectors (items) by retrieving the cosine of the angle between the two items to determine whether or not they’re alike. The formula for calculating cosine similarity is below with A and B representing the respective vectors. The final answer of the formula will produce a figure between 1 and 1; -1 signifies that the items are not at all similar with 1 stating a strong similarity prediction.

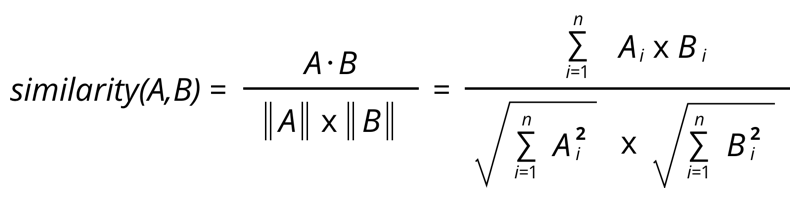


Figure 9: Formula for calculating cosine similarity

The major drawback of content-based filtering is that there is rarely adequate information to determine what items a user prefers and doesn’t prefer – leading to a very broad final output.

### Measuring Recommender System Accuracy

When creating a recommender system, it is common to implement several versions of experimentation to find the ideal technique for the intended dataset. A crucial area of choosing the right technique is to measure the accuracy of your recommender system. This is achieved using two measures: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). A few years ago, Netflix organised a challenge (the “Netflix prize”) where the goal was to produce a recommender system that performs better than its own algorithm with a prize of 1 million dollars to win. The winner was chosen on the basis that a team could achieve the lowest RMSE, influencing those competing to tweak their recommender systems towards that common objective. This ideology of creating recommender systems to achieve low RMSE scores has carried through to present day. However, researchers claim that user testing may be a better approach in developing more successful recommender systems.

## User Experience and Recommender Systems

The main reason we see recommender systems in high demand in today’s market is down to the significant increase they cause in sales. Algorithms that focus on user activity are implemented to generate accurate and relevant recommendations, which in turn causes increased sales figures as the user is offered items they are expected to enjoy and avail of. There are, of course, alternate factors that can alter the recommendation process and influence the user experience. It isn’t true that using a recommender system is only essential for increasing product views; good and efficient recommendation engines will offer a more satisfying experience for the user throughout their visit to a particular website and this also needs to be taken into account when picking the ideal algorithm for a user.

## Research Conclusion

In Conclusion, the aim of the Research document was to assist the design and development of the music-based recommender system. By following the available resources and conducting in-depth research in to how the common recommender system functions, it has now become clear what kind of drawbacks may feature during the development. Firstly, it has become apparent that choosing the correct recommendation system will be difficult, whilst Collaborative based filtering seems to be a more favored recommender engine for more accurate results, it may prove difficult to implement such an engine into the application, content-based filtering has proven to be slightly favorable during initial prototypes. This could prove to be less effective if taken forward as recommendations may not be as fine-tuned and as a result leave the user feeling unsatisfied with what they have been presented with. Secondly, the question of what kind of data to use has also been raised. Whilst initially, the project will feature a dataset, the question of implementing an API has been discussed. Whether this can/will be added to the application has yet to be seen but will be a feature that will be looked further in to for definite. Nevertheless, understanding the expectations the user wishes to see will be critical in the implementation and design process.