MSc in Intelligent systems

Smart Recommendation System for Music Using Data Analytics Report

Student Id: 2664093
Email: P2664093@my365.dmu.ac.uk

Supervisor name: Dr. Ioannis Kypraios

Module info: IMAT5314_2122_519
School of Computer Science and Informatics
Faculty of Computing, Engineering and Media (CEM)

Date: 02 September 2022

ABSTRACT

This report focuses on designing a smart recommendation Web application for music using a cloud-client (server) architecture. Google cloud Firestore database has been used to store the data and atom IDE was used to script in JavaScript. Implementation of data analytics algorithms with machine learning i.e., Distance classifiers such as Euclidean Distance, Manhattan Distance and RBF to perform the recommendations for the user. The recommendations will be viewed in a PDF format to the user.

Keywords – Music recommendation systems, Data analytics for recommendation applications, Web application and machine learning

Table of Contents

Chapter 1

Introduction

1.1 Background

Information is multiplying every day. Information overload has become a huge issue. All the technologies and internet especially, have made it all easily accessible. Although, streaming the relevant information among the entire data remained as a hassle to the users, recommendation systems (RS) addressed the issue of information overload by helping the user stream related content. RS are developed to suggest relevant content to the users based on their interests by filtering the data. This concept first appeared in 1979, when it was implemented in a system named Grundy [1] (Computer Science, 2020), where a computer-based librarian gave suggestions on 'books to read' to the users. Followed by the Tapestry launch, by the Xerox Palo Alto Research Centre in 1992 [2] (Boehmer, Jung and Wash, n.d.), to address an issue - overflow of emails. The tapestry was designed to filter the emails with topics marked as relevant by the user. In recent, where many activities have become online and digitalized, recommendation systems are widely deployed for its ability to attract customers with an improved on-site experience by using their data history to provide suggestions which would more likely match their preferences. Some examples are search engines, shopping websites, music and video streaming platforms, social media, academic courses, mobile applications and on the web. Recommendation systems satisfyingly meet the customer's need in a practical and efficient approach.

1.2 Impact of Recommendation Systems today

Music is cultural and universal. Besides the fact that its data is increasing every day, the demand of users has been increasing as well. Every single customer has a different playlist with different taste in music. With music becoming the background of today's daily life, an extensive need for music recommendation system was met. Various filtering procedures and algorithms have been implemented to develop a music recommender system. The music recommendations have become stronger and smarter throughout the years with the techniques introduced and implemented. With of pandemic changing the entire scenario of shopping in stores in 2020, customers have made a shift to shopping online from shopping in store which has resulting in the demand of recommendation systems to be emerged. Majority of companies are willing to emerge technology and Artificial intelligence into their business to maximize profits and attract customers with hassle-free services. In 2020, the engine market size of global recommendations was valued at 1.77 billion USD and is predicted to grow at a compound growth rate of 33% for the following 7 years [14]. Spotify, Amazon Prime, Netflix, YouTube, Facebook, Instagram, Tik Tok etc. are some example of successful business cases where recommender systems have outperformed by suggesting more of what could already be of user's interest leading to an inward spiral of similar influences. It is estimated that around 35% of the profits to Amazon, 80% of watch time at Netflix, 2.3 billion music hours discovered weekly at Spotify, typical 52 mins a day Tik Tok viewed by user are some of the significant impacts of recommendation systems in businesses.

1.3 Objectives and deliverables

By having great interest in music and by observing the impact of recommendation systems in today's world, the scope of this project is to design a web application using a cloud-client (server) architecture using machine learning to implement different data analytics i.e., different distant classifiers discussed further in chapter 4. The user will be able to view the recommendations based on selected preferences in a PDF format. There are 5 objectives (mentioned below) and 6 deliverables to be delivered on the completion of the project, the tasks and deliverables are presented in the table 1. The timeline of this project is visualized in Gantt chart, given in figure 1.

Objective 1: Completion of state-of-the-art literature review (chapter2) and the data analytics algorithms for smart recommendation systems.
Objective 2: Design of the web app layout.
Objective 3: Implementation of the Data analytics algorithms with machine learning i.e., Distance Classifiers such as Euclidean Distance, Manhattan Distance, and the RBF
Objective 4: Testing and Use Cases.
Objective 5: Critical analysis of your system and suggestions on limitations and improvements.

Deliverable 1: To deliver the Ethical review form with completion
Deliverable 2: To complete the writing up of chapters 1, 2, and 3
Deliverable 3: To complete the development of Web Application Layout
Deliverable 4: To complete the implementation of basic data analytics for calculating the 4 Emotion categories and the three distance classifiers for similarity measurement
Deliverable 5: Results and final write up of the thesis
Deliverable 6: Submission of MSc thesis

Table 1: Project time plan

Tasks (as WPs) & Milestones (M)	Deliverables (D)	Start date	End date	Duration
WP1: Fill in the ethics review form		29-May	30-May	1
M1: Ethics review form approval	**D1:** Ethical review			
WP2: Writing up of chapter 1: Introduction		01-Jun	15-Jun	15
WP3: Writing up of chapter 3: Methodology		12-Jun	30-Jun	18
WP4: Writing up of chapter 2: Literature review and creation of Firestore database.		25-Jun	16-Jul	20
M2: Completion of chapter 1, 2 & 3 along with Gantt chart	**D2:** Chapter 2: Literature review			
WP5: Design of the web app layout		16-Jul	30-Jul	15
	D3: Web page layout			
WP6: Implementation of data analytics algorithms with the distance classifiers		01-Aug	15-Aug	15
	D4: Data analytics code			
WP7: Test cases with the autogenerated smart recommendation Report		15-Aug	20-Aug	5
M3: Finishing up the design				
WP8: First finalized draft of the MSc thesis.		17-Aug	29-Aug	12
M4: MSc thesis	**D5:** Results			
Critical analysis of the system and proof reading		29-Aug	02-Sep	5
M5: Submission @ 12:00PM	**D6:** MSc thesis	02-Sep		1

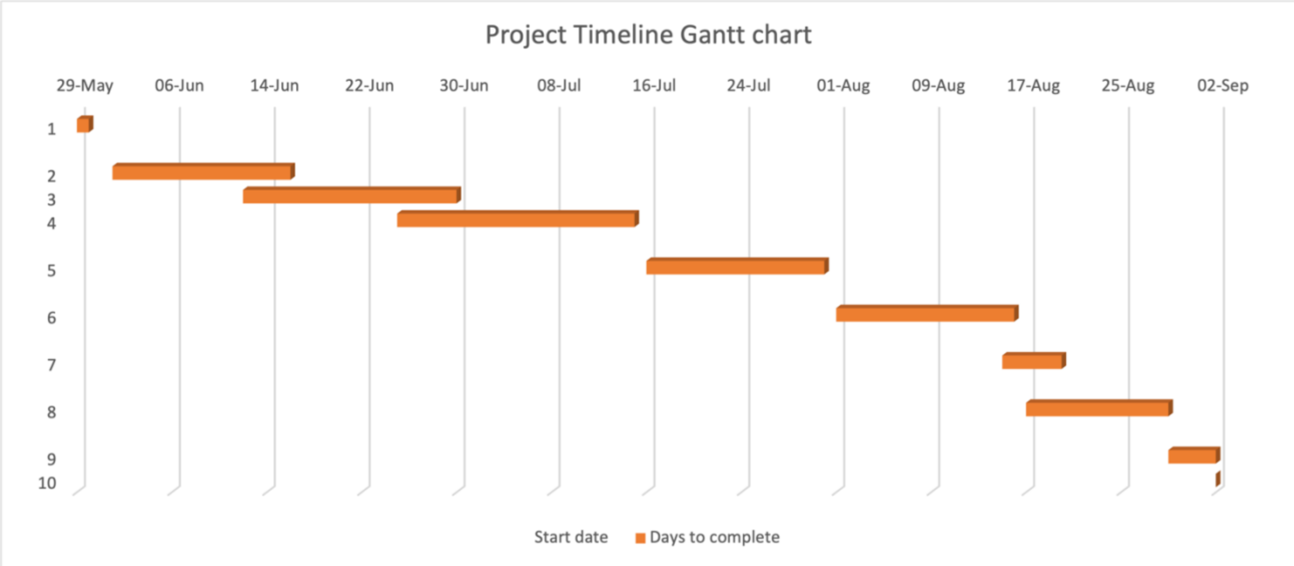


Figure 1: Timeplan - Gantt Chart

Chapter 2

Literature Review

This section consists of related works reviewed on 4 different analyses i.e., firstly on Emotion basis, secondly on Genre analysis, thirdly on social network analysis, fourthly on data analytics and different machine leaning approaches, and lastly on work related to Music recommendation systems, explained in order below.

2.1 Emotion Analysis

Human emotions can be a potential context to improve the quality of music recommendations but acquiring the emotions remained to be a challenge so far. Taking this into account, the authors (Shuiguang D, Dongjing W, Xitong L and Guandong X, 2015) [1] proposed a way to extract user emotions during different times and at various granularity levels by utilizing user's microblogs (which is proven to be improve the recommendations more conducive), which later integrates the three elements i.e., music, user and the emotions extracted for recommendations. With smartphones and microblog services becoming prevalent, the assumption was that microblogs can be used to process emotion vectors for user preference prediction. Several emotion-aware techniques which generate music recommendations were developed in this experiment. The tradition Collaborative filtering user-based method was deployed for similarity computation along with Bayesian for personalized ranking, to evaluate and compare the results. The results proved that methods considering emotions have outperformed the existing methods and have given significant results. The authors also state the model's limitations and provide suggestions for future experiments. The authors (Rosa, Rodriguez and Bressan, 2015) [2] worked on developing a music recommendation system using a technique of Text Analytics from Natural language processing called "Enhanced Sentiment Metric" which was associated with a correction factor which depends on each user's profile. The correction Factor was discovered through subjective tests, the factor was formulated to fit and generate desired recommendations. The authors have proven that the technique of sentiment analysis together with a weight for influencing the suggested readings improved the performance of the user recommendation. The factors discovered were extracted from the wider daily use of social networks. Lexicon-based techniques and recommendations from Natural Language Processing (NLP) were used to extract emotions within words. Further, by utilising the crowdsourcing method with remote users, the outcome of the results was evaluated with a rating of 91% satisfaction. The authors demonstrated that those results outperformed the previously conducted random experiments.

The authors (Bodarwé, Noack and Jean-Jacques, 2015) [3] based on feature extraction, automatic classification and emotional computing proposed a music recommendation system which generates suggestions based on the song's emotion. The approach was to implement a system where a classifier (in this project they used Tellegen-Watson-Clark model's simplified version to distinguish between emotions more easily) trained with audio extract features, determines the emotion of the song. A Naive Bayes classifier was deployed as it showed better results that other classifier such as Support Vector Machine, Logistic Regression, Decision Tree especially when classifying music based on emotions. A 10-fold cross validation technique was used to evaluate the classifier. The accuracy yielded was 53.07% which only improved when other labels such as lyrics, genre information were included. The authors claim that classification of music through emptions was a challenging objective and would need more research to be carried to be able to achieve it. The authors (Su and Zhou, 2022) [4] have developed a Machine Learning model to predict the intrinsic relationship between human emotions and music. To develop a genre-diverse music playlist, the author carried out a heuristic clustering analysis on large music data. 8 machine learning methods were implemented to compare the different outcomes (1) Multiple linear regression (2) Partial least squares regression (3) Back propagation neural networks (4) Support vector machine (5) Least-squares support vector machines (6) Random Forest (7) gaussian process and (8) Radial basis function neural networks. The project was explained in 3 steps (1) data pre-processing (2) Partition of the diverse library (3) Model Validation. It was discovered that nonlinear machine learning methods (6 out of 8) were a much better fit for this project than the linear machine learning methods (2 out of 8). The model was shown to have a good predictability.

2.2 Genre Analysis

The author (Singhania, 2020) [5] published an article explaining how to understand different music genres and their themes for developing a music recommendation application. The author was inspired by the success of applications such as Spotify. Amazon previously had built a model to predict the genre using the lyrics of a song. Thus, they conducted some analysis using the available data for classifying the genres of 100,000 albums using a graph. The author applied a distance measurement classifier called the 'Jaccard similarity' for identifying all the matched songs to the selected and entered user content. The author found that the conducted results from the model were within a 73% accuracy and 74% recall. Other corrections and modifications were made to improve the accuracy up to 80% as was shown on the confusion matrix. The author (Boyd, 2019) [6] conducts a detailed analysis of Spotify and how its algorithm using natural language processing and neural networks, succeeded in performing excellent recommendations with other recommender systems in the market. The author talks in detail about the techniques used in developing the model. The figures such as total revenue, advertisement support, subscriptions, and all the costs to company as considered. Its reach in the market along with competitors were discussed. A detailed analysis of all the existing data was gathered. Major questions about Spotify were answered. This report succeeding in giving a detailed report of Spotify and how its algorithm performs, and the factors considered to influence the suggestions. The author (Dwivedi, 2019) [7] published an article explored over the architectures combining the RNNs and CNNs for classification of audio music clips into eight genres along with filter activations visualisation in different CNN layer. The dataset available on GitHub was used. Along with Convolutional Recurrent model inspired from deep sound's work, a parallel CNN-RNN model inspired by Lin fen and Liu's work was deployed as well. The keras Visualisation package was used for visualisation activation. The validation loss was around 51%. Both models have shown similar accuracy although parallel CNN-RNN has performed better in some genres.

2.3 Social Network Analysis

The advancements in the technology-music field the growth of music recommendation models is rapid. With social media and networking growing the normal routine, social data can be used to extract user preference information to improve recommendations. The authors (Ting IH and Yu PL, 2016) [8] have conducted a series of experiments (1) to measure the music recommendation's rating using the social data information (2) to study the impact of different social data on recommendations evaluated by user ratings. Facebook (the like pages, fan pages timelines and feeds) was used to extract different types of social data of 53 users for this experiment. GEMS emotional tags proposed by Zenter, was deployed along with cosine similarity to measure similarity. Music liked fan pages contributed the most to improving recommendations compared to other kinds of social data extracted from Facebook. This experiment resulted in proving that the social data (categories used individually or combined) provide improved recommendations compared to random selection. The authors (Knees P, Schedl M, 2012) [9] published a survey article with a detailed overview of methods to identify the music similarity estimation and context data-based music recommendations. The survey article explains the characteristics of the given context-based measures and shares their opinion. Before examining, the authors had a detailed examination do data available up to date. Different methods of context-based data for each artist and music track. Different approaches to implement similarity measures based on the knowledge fed as input was briefly summarized. Three important types of context-based approaches namely (1) Co-occurrence based approach (peer to peer networks, replying on playlist) (2) Text retrieval-based approach (tags, lyrics) and (3) based on data from ratings and history approach. The exploitation of this information and let the authors to some conclusions. Approach based on texts such as lyrics or general we search engines instead of audio-based approach are not to be considered as community dependent. Although avoidance of community dependency could lead to scarcity of inputs to feed for recommendations in the future.

2.4 Data analytics Analysis

Conventional Music recommender systems are user-track relationship based and recommend songs based on intrinsic factor ignoring the user-contextual factors which could significantly improve the recommendations. To overcome this gap, the authors (Wang R, Ma X, Jiang C, Ye Y and Zhang Y, 2020) [10] proposed HIN-MRS i.e., Heterogeneous information network-based music recommendation system. This method considers all the factors from all aspect that are extrinsic, internal, and contextual and the heterogenous connection between the song information. The approach was explained in 2 steps (1) to obtain textual data to obtain user's preferences to provide a contextual factor related topic and (2) based on the topic determined, a small-scale HIN songs and used to generate recommendations in a graph-based algorithm. The objective of the experiment was achieved, topic extraction can identify the emotions and improve recommendations effectively. Additionally, an innovative means to automatically capture weights of relations in a network (heterogenous) was proposed. To bridge existing the research gap which was to create a personalized smart product service system (PSS) which would benefit everyone in the process, the authors (Chiu M, Huang J, Gupta S and Akman G, 2021) [11] proposed a method which included (1) Analysing user-provided data using unsupervised natural language processing (2) providing personalized solution or users integrated deep learning techniques in this recommender system. This system was to allow enterprises to create their personalized PSS and to enable them to modify their existing business. The authors explained their method in four steps. Firstly, from the existent PSS to identify the requirements of user. Secondly, explored key data to meet the user needs. Thirdly, stable models were constructed optimize the current PSS and finally to develop smart solutions as a result. This research succeeding in the experiment conducted and has given smart solutions. Suggestions were made for further research in this area considering the drawbacks of this project. With recent research proving that a conceptually simple method called "Nearest Neighbours" schemes can be a viable technique for content recommendations, the authors (Ludewig M, Kamehkhosh I, Landia N and Jannach D, 2018) [12] proposed a hybrid technique for generating the next-track recommendation developed in the context of ACM RecSys 2018 Challenge. A few nearest neighbour techniques were combined along with a standard matrix factorization algorithm and some small heuristics were deployed for the development of this system. Item-based collaborative filtering was used to consider all the data. The session based KNN to consider the entire playlist to find out the song with highest similarity. As a last tract-based component of their hybrid approach, metric factorization was included. This project proves that KNN is the most suitable technique for finding similarity. However, combining different techniques led to generating improved results. All the model's individual strength is leveraged by the hybrid approach. The authors (Schedl and Kepler, 2013) [13] investigated three categories of music recommendation systems i.e., music content, music context and user context. They studied different approaches in addressing: (i) geospatial music recommendation from microblogs, (ii) user-aware music playlist generation on smart phones, and (iii) matching places of interest and music. They described an intelligent music player for mobiles called the Mobile Music Genius (MMG) which automatically adapted the current playlist to the user context. The authors used twitter microblogs to develop their user aware MMG linked with geospatial information. MMG could dynamically adapt the music playlist by continuously monitoring a wide variety of user context data while interacting with it. The authors collected user annotations for 25 places of interest and for 123 music tracks using an interface on their MMG. Then, they experimented in linking a specific place with a selected music piece using five approaches: (1) a knowledge-based, (2) a user tag-based, (3) a music auto-tagging, (4) a combination of (1) and (3), and (5) a simple personalized baseline. The authors argued that their developed MMG achieved a higher number of matches returned and an increased user satisfaction.

With existing content-based music recommendation systems being a two-stage approach, which is to firstly collect all the audio content features traditionally and secondly to predict user's preference. The authors (Wang and Wang, 2014) [14] introduced a unified two step approach to predict user preference in a content-based method using novel deep belief network models as recent findings have proved that it can be possible. To integrate the automatically learnt concepts and the collaborative filtering, the authors presented a Hybrid method proved to be efficient. The content-based model was based on Matrix Factorization. The hybrid model mainly uses on Data Fusion to develop a unified model. The model where mathematically represented explaining the implementation. The Root Mean Square Error was the evaluation metric deployed. The proposed model generated satisfying results compared with the current deep belief models deployed without relying on collaborative filtering. The hybrid model introduced has outperformed the existing model by also improving the CF.

The authors (Vairavasundaram, Varadharajan, Ravi and Vairavasundaram, 2015) [15] presented an overview of the tag recommender system and factors that are impacting the current tag recommendations system. They further proposed that spreading activation algorithm to use for the study of constructed topic for efficient recommendations of tag. The paper was explained in seven main sections with a detailed exploration of history and current situation of tag recommendations. Precision, F-measures, and recall were some common metrics kept as a benchmark of the derived results. Other datasets I.e., BibSonomy and Delicious were used as examples of tags extracted from the blog. The Folksonomy dataset used helped in finding relevant tags. Their process proves to use topic ontology approach for effective recommendations of tags. The tags recommended were accurate and valuably retrieved. The authors (Zhao and Liu, 2019) [16] published an article on a study proposing APIUaaS - API Usage as a Service, for facilitating API usage, to allow infrastructures built for recommending the correct API code demonstrations on partially automated data analytics, a referencing architecture - APIUaaS is used. This study examined the gaps of current API usage recommendation approaches and has presented a solution and demand for making API into services. Firstly, the current perspective has been considered for literature and they sketched the ideal user perspective by enabling data mining and code recommenders for API programming. With the analysis of existing recommender, an APIUaaS was inductively designed. API makes use of mining, ranking, and clustering algorithms enabled to generate recommendations. The core concerns being six global-level architectural and five logical layers, were satisfied with the results of this project. The results showed that blueprints could be provided by the APIUaaS for the developers. The authors (Xiong, Lu, Bing, Hang and Wu, 2018) [17] proposed a web-based smart service recommendation approach through representation learning utilizing deep learning techniques. Large scale experiments were performed over real-world dataset to validate their approach. Their approach consisted of 4 components (1) NLP pipeline to pre-process the input (2) Word modelling with character to word model - C2W based on bi-directional LSTMs (3) Language modelling of semantic relationship and (4) Embedding-based model recommendations. The accuracy of their approach and their embedding model's performance and accuracy. Experiments were designed to evaluate their approach throughout experiment. The results were quite promising, but some challenges remain to exist and suggestion for further studies were shared. The authors (Tsaku and Kosaraju, 2019) [18] introduced the three recommendation systems which are

Knowledge-based, content-based and collaborative filtering systems to suggest a new method for existing customers that integrated new user's domain knowledge into a recommendation system based on machine learning. They also proposed an online/offline strategy for evaluation to reduce inaccuracy in recommendations for all users. They also expanded on the ideas of Matrix factorisation and SVD and FunkSVD. A movie dataset was used to implement each recommendation system along with the proposed model for comparison of results. Firstly, to generate recommendations, they integrated the domain knowledge into the matrix resolution machine learning recommender. To enhance the recommendations for users with no history, content-based recommendations were merged with the resulting matrix resolution recommender. The results show that this method has challenges to still address with but has proved that the hybrid method is quite effective.

2.5 Music Recommendation Systems

To address the cold-start problem on how to suggest a new artist without prior knowledge of the user, the authors (Oramas, Nieto, Sordo and Serra, 2017) [19] by combining audio and text information with feedback data using Deep neural networks dividing the method in three parts. The approach was to first learn from biographies by aggregating the factors and secondly from available user feedback data. at last, they combine all the embeddings (learnings) into a multimodal network. Mathematically represented as Fs =As ∪Ts. Where F is a featured list. A is the artist and T is for the tracks in their album i.e., the aggregation of artist and their tracks. Results prove that learning artist featured tracks embeddings separately from the feedback data benefits the model. This model achieved better results that plain texts and audios approach. The authors (Lee W., Chen C, Huang J and Laing J, 2017) [20] recognizing the contribution of contextual information worked together to explore the purpose of using a smartphone to identify physical activities. This information is used as a feed to generate music streaming recommendation. With each user having unique need to be meet and to make their suitable music choosing and easy task were to two main issues to be focused on with this project. The framework consisted of a client-server architecture; cloud based. The embedded acceleration sensor was deployed to collect activity recognition data and created an interactive system to enable physical interaction between the machine and user. A hybrid of content-based and collaborative recommendation modelling methods is used, formulated the procedure, and represented in diagrams. From all the tests and results data, accuracy from different methods ranged from 66.4% to 83.4% for human activity recognition (HAR) and 71.5% to 77.6% for music streaming recommendations. The results proved that this method was efficient and further studies are carried on. The author (Si, 2021) [21] noted the significance of the sound of the piano together with the attributes of Deep Neural Networks and developed a piano music recommendation algorithm. He used 4 concepts of Convolutional Neural Networks (CNN) for this purpose (1) a local receptive field for representing the range of neurons in the network; (2) a feature mapping with two consecutive layers of shared weights or shared biases; (3) a convolutional kernel and (4) pooling algorithm for simplifying the output. Those four steps were represented as a formula with all the variables defined. The author concluded that his piano music recommendation algorithm had produced good performance due to the deep learning algorithms. The authors (Kim, Kim and Kim, 2018) [22] developed a mobile-based music recommendation system with two modules: (1) a module for recognizing human activities using a deep recurrent neural network and (2) a module for recommending tempo-oriented music accordingly using a modulation spectrum and a sequence classifier. Accelerometer sensor data from smartphones were used to develop an effective user recommendation system based on the human activity recognition (HAR). The whole experiment was carried out in 4 steps. (1) A mobile music player (tempo-based) was developed using a mobile-phone accelerometer (2) Six activities were classified using data augmentation for a deep residual bi-directional gated recurrent neural network (3) An ensemble method was implemented for generating recommendations with accuracy. (4) The techniques were integrated for improving the performance. Thus, the results for the tempo-oriented music study demonstrated high accuracy recommendations. The developed mobile-based music recommendation system achieved accurate results when compared with the manually annotated recommendations. The authors (Duo X, Yonghui and Weihui, 2019) [23] considering the personalized requirements for the recommendations of cloud music service to satisfy the audience needs and efficiency while streaming music, conducted a neural experiment to record and analyze the concerned factors of these services to study the model to describe and figure out user's mental preferences. Based this study they proposed a smart recommender which embedded rational and emotional intelligence for personalized cloud music services. The authors say that Mental models can help guide people in activity conduction while facing the unknown environment. The electroencephalograph (device used to analyse audience's satisfaction and concerns regarding the system) used here are Oddball, Go-Nogo and N-back along with traditional methods like surveys, questionnaires interviews to obtain the user's cognitive data of music. The experiments were carried out on a crowd of 48. The results were investigated and displayed in a pie chart. To address the issue of choosing suitable promotions from the great variety of telecom offers promotions for customers, the authors (Hu W, Yu C, Tang S, Chen Y and Hsu W, 2018) [24] proposed a promotion recommendation method and system based on random forest (by adopting a base classifier) to be able to analyse users' profiles and history of mobile data usage for market information to be obtained. This information could guide the front-liners to make accurate strategies of marketing and make suitable recommendation promotions. 500,000 mobile usage data was used to test the cases. Using the random forest's tree green process, they obtained the importance of all nine-customer feature in this project. The results after modifications from the experiment reflected with an accuracy of 93.4% gaining an advantage over the remaining three well known classification algorithms i.e., Support Vector Machine, Multi-Layer Perceptron and Convolution Neural Network.

To improve the user's interaction with recommendation systems and to deal information overload, the author (Zhao, 2021) [25] proposed a machine learning-based algorithm for recommendation of pop songs. With the traditional recommendation system having shortcomings such as cold-start, data scarcity etc. the author studied and redesigned the traditional recommender system by integration the machine learning algorithms. The machine learning techniques included Artificial Neural networks, K-Nearest neighbours and computer learning prediction algorithm. This system focuses mainly on historical data of the user to fetch for songs to recommend. The short-term prediction success and recall seemed to improve with a larger dataset, but the results in the end were not very satisfying, and the author states the concerning factors to be considered in the upcoming experiments in this field. The authors (Hong D, Yang and Dong Q, 2020) [26] published an article which proposed a music recommendation's novel policy – NRRS (Nonintrusive-Sensing and Reinforcement-Learning Based Recommendation Systems) by integrating streams from previous research. To sense, learn and adapt the user's present preferences based on wireless sensing in real time while the user is listening to music was the specific task to be addressed while developing this system. The NRRS architecture as designed with a goal to find and improve the terms in time over which song should be played as the upcoming track. A sequence of songs liked by the user is created by the NRRS recommender, each song's completion will start reading reactions of the user through wireless sensing. This feedback will be used to adaptively improve the playlist in no time. Finally, it was significantly clear that the NRRS recommender has outperformed with its results by demonstrating that it can effectively achieve desirable recommendations. Identifying the impact of integrating the listener's cultural background information for building a music recommendation system, the authors (Zangerla E, Pichl and Schedl, 2018) [27] proposed a novel approach to model users based on their cultural background derived from user's location and musical preferences which is the acoustic qualities of the user's playlist. They state that by building a system with this information from the user will help the system to identify cultural listening patterns. The model proposed exploits these listening pattens to generate recommendations by integrating two factors. The popular XGBoost clarifier was utilized for a scalable boosting approach. With evaluation and further testing with correction, the approach displays that the country of user can not

be a proxy to describe cultural aspects and hence only the recommendations with musical preferences were more accurate that geo-based recommendations for cultural music.
To overcome the issue of slow convergence in recommendation systems, the authors (Pereira BL, Santos R and Chaves, 2022) [28] proposed a novel online-learning ranks approach. A session-aware exploitation component was proposed to overcome local minima. The authors' contribution was two-fold (1) an online learning approach was proposed for recommending music which ignores exploring unpromising areas and (2) the authors thoroughly evaluated their proposed approach regarding its effectiveness and efficiency in comparison to the contextual bandit approaches. Null Space Counterfactual Dealing Bandits (NSCDB), the proposed approach, was successfully developed. The 22-dimensional representation was used for feature engineering. The outputs were enhanced by enabling the said factors. All results were with a 95% interval for the computation made using 1000 samples. Future works are intended to be carried out by the authors in this field further run this topic. With an idea that a successful recommender system must explore the user's music preferences and equally exploit the extracted information to generate recommendations has served as a motive to the authors (Xinxi W, Yi, Hsu and Ye, 2014) [29] to propose a new approach to recommend music by formulating a reinforcement learning task from the exploration and exploitation trade-off (which has two baselines: Random and Greedy). Bayesian models along with Variational inference algorithm and some enhanced techniques such as piecewise-linear approximation was used to learn the user preferences. A unified model that recommended music and as well as generated playlists was developed. Out of the 3 factors and 4 approaches, 6 recommendation algorithms were evaluated in this experiment. The simulation's result was shown to be effective and accurate using this bandit approach to avoid cold-start problems. The effectiveness of such combinations was intended to be studied in future works. The authors (Knees P and Schedl M, 2015) [30] provide an elaborated tutorial to the Music Information Retrieval (MIR) field. The objective was to review approaches that obtain features from 3 central data source factors and combinations (i.e., the music content, the music context and the users and their context) and demonstrate how this information can contribute to improve recommendations, measuring similarity, music description and indexing. Popular music applications deploying these techniques were demonstrated. Automatic music accompaniment, identification of music, retrieval, search, and other related topics were briefly discussed. This tutorial article gives a complete picture of MIR in 3 sections (i) Content-based MIR (ii) Contextual Music Similarity, Indexing and retrieval (iii) Collaborative Music similarity and recommendations. The tutorial's outline structure was designed well and presented.

Chapter 3

Methodology

3.1 Research methodology

Research methodology is a systematic approach on how to achieve a new conclusion to a research problem. It explains how you approach a problem and gives a scientific and commercial value. In simple words, it is about "HOW" the researcher designs their study systematically to ensure reliable and valid results addressing the project's aim and objectives.

Research methodology can further be classified into two types based on their models.

- **Quantitative research methodology**: Quantitative research methodology deals with models which have variables that can be measured and counted i.e., numerical data, statistics, etc., therefore this method is objective. These models can be systematically tested or proved.
- **Qualitative research methodology:** Qualitative research methodology begins with observation and exploration of the data and its concepts to be able to interpret them for extracting parameters and building the model. This method deals with data that is descriptive and language related, therefore this method is subjective.

Irrespective of the types, quantitative research methodology and qualitative research methodology can be approached in three ways:

- **Top-Down research methodology:** In the Top-Down approach the topic start from existing theories and concepts, then it narrows down to the design and development of the model. Figure 2 provided by Dr. Ioannis Kypraios (Supervisor).

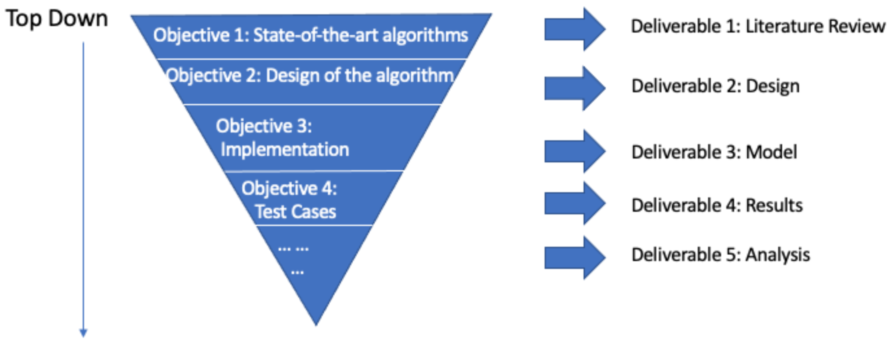


Figure 2: Top-Down research methodology visualization

- **Bottom-Up research methodology:** In the Bottom-Up approach the topic starts from interpretation and observation of the data and later generalizes in the topic extracting the parameters to build the model. Figure 3 provided by Dr. Ioannis Kypraios (Supervisor).

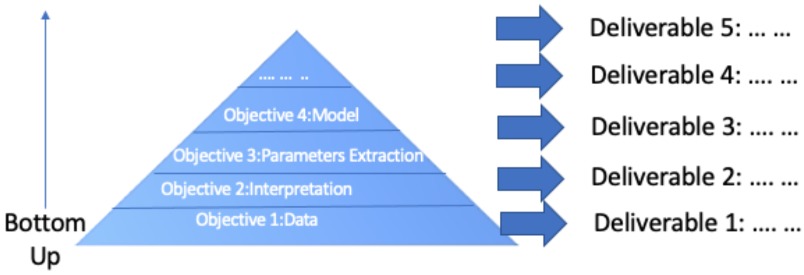


Figure 3: Bottom-Up research methodology visualization

- **Mixed research methodology:** This is usually a combination of Top-Down and Bottom-Up approaches, it is mostly used in PhD research.

3.2 Implementation methodology

Implementation methodology is a systematic guidance showing us the techniques need to be followed to achieve the deliverables. In simple words, it contains all the methods to apply for achieving the deliverables.

3.3 Agile methodology

In this method, the entire project is managed by breaking it down into several phases involving continuous collaboration and improvement at every phase. As user's journey, need and preferences from the software cannot be accurately predicted throughout, Agile methodology was designed to address this issue. In simple words, the research methodology and implementation methodology are formalized into the full software development cycle. The Agile Methodology is the standardization of the full software cycle. With the implementation of work for specific period, which is predefined with intervals, one sprint comes to completion. Each 'sprint' (more like each iteration) makes sure that the software deliverables, needs, and changes are addressed as planned [11].

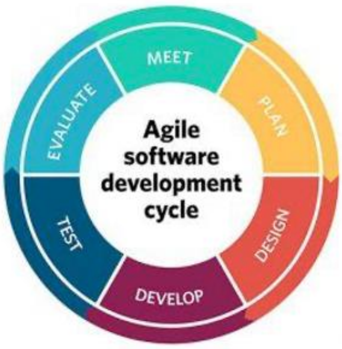


Figure 4: Agile Methodology – full software cycle and sprints.

On a general basis, a recommender system's procedure majorly has three components [3] (Roy, 2020) -

- Personalized selection of data on suggestions: Generating small subsets of the entire data based on customers preferences
- Scoring stage: Standardize the generated subsets and assign a score to each content
- Re-Ranking stage: After the scoring stage, this stage considers additional constraints to confirm the final ranking.

The crucial step, personalized selection of data to be on suggestions can be done in three ways [3] (Roy, 2020):

- **Content-based filtering system:** Content based filtering recommends the content based on the customer's reaction on the given content. If the reaction is positive, the similarity metrics is calculated between the remaining data with highest matching features to the user's previously positively reacted records in the RS and the top few items are put forward in suggestions. The data needs to be assigned with features to be able to compare to filter out. Other user's data is not required in content-based filtering system. For example, a movie recommender system with a content-based filtering system, would analyze the similarities such as authors, titles, genres, etc. of the movies previously rated or liked by the user and then recommend the one with highest similarities.
- **Collaborative filtering system:** In collaborative filtering system, the content need not have assigned features. This recommender system also uses other user's data and history of likes and dislikes to predict and bring in suggestions to a new user. Instead of calculating similarity metrics, it embeds the user and items in the same embedding space. Feedback and patterns of choice is collected from all users to use them for generating recommendations. For example, a movie recommender system with a collaborative filtering system, would analyze other users who have positively rated some movies including the one liked by the current user and suggest the other movies which were also liked by the other user. This filtering system type is often referred to as "people to people correlation".
- **Hybrid filtering system:** Hybrid filtering system is a combination of two or more filtering systems basically combined to overcome the disadvantages of one another. For example, in collaborative filtering system the recommendations are made from similar user's preference which means that the new items which are not rated by anyone often don't make it to the suggestions., but with content-based filtering system combined, the suggestion are made with the content that share high similarity and not just user rated content. Most RS are developed with hybrid filtering system due to their least limitations and for the convenience of combining the available filtering systems.

According to this article (Marty Shuttleworth, 2008), some basic steps have been followed from ancient times irrespective of the scientific discipline. They include (1) Narrowing down to the research area and proposing a realistic hypothesis by considering the available resources (2) Designing the experiment and the stages in it for testing and evaluation with statistical tests in mind (3) Observing the results along with the findings from raw data and recording it – mid stage of steps in scientific method (4) Statistically analysing the performance of the data and (5) Drawing a conclusion - proving or disproving the point followed by publishing it for others to continue research in this area areas. These simple steps guide us to carry out research scientifically. Figure 5 below gives us a visual representation of the steps of the scientific method [12].

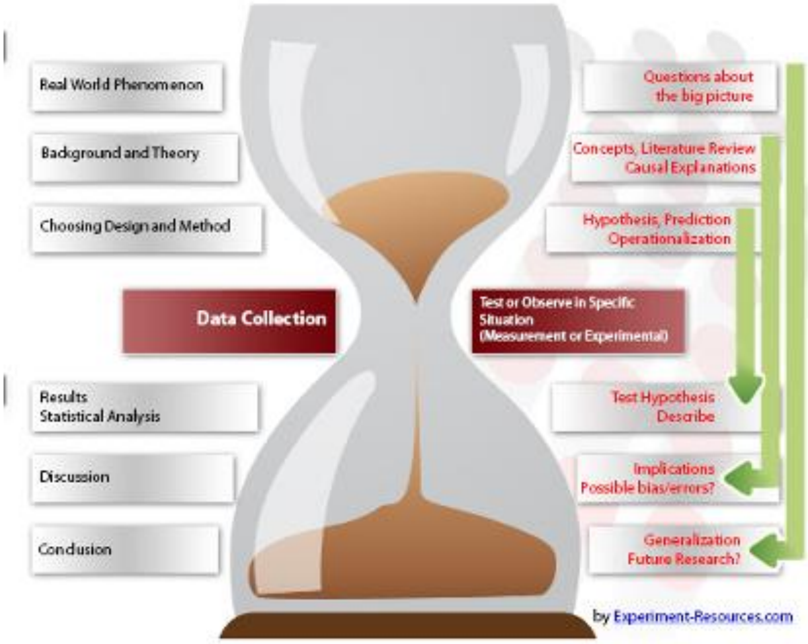


Figure 5: Steps of a scientific method [12]

3.4 Data and Research

Data and research can be classified in two types (1) Primary and (2) Secondary, as explained in table 2. For this project, we have generated 50 music albums using generatedata.com (explained in detail in chapter 4).

Table 2: Primary and secondary classification

	Data	Research
Primary	The data gathered for the first time by the researcher themselves by observing a new phenomenon or researching never-before researched areas. It is more reliable as it is new data.	Original research conducted by the author(s), specific to their research topic. It can be done with primary data or secondary data.
Secondary	The data which was previously collected by someone and used in the current research are called secondary data.	This research topic has already been approached before but is being performed again to derive new conclusions.

3.5 Smart Music Recommendation System – VIBE

This project, “Smart recommendation system for Music with data analytics” is a *quantitative research methodology with a Top-Down* where all the deliverables are achieved. For this project, we will be conducting *secondary research with secondary data* openly available (in this case generated from the website GenerateData.com). This web application was named “VIBE” and given a logo from the famous cat meme for entertainment purposes. Cat image downloaded from google.

The figure 6 below was created referencing from figure 2. It demonstrates the Top-Down approach of this project with the objectives and deliverables.

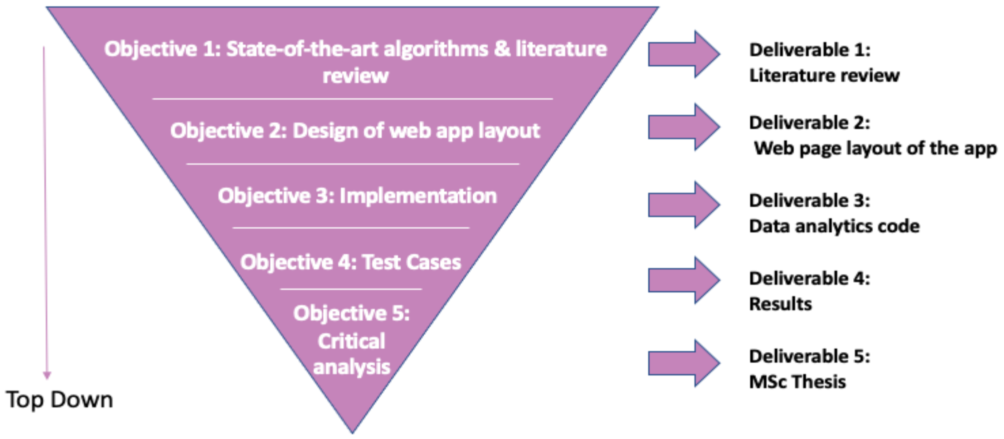


Figure 6: Top-down approach visualization of this project

Chapter 4

Design of the Web App

4.1 Web Application Architecture

Web application is not really a website but a program that performs on a browser. “Web application architecture is a mechanism that determines how application components communicate with each other. Or, in other words, the way the client and the server are connected is established by web application architecture” [9] (AltexSoft, 2019). The device used to browse with and access the web application is called the client. The other end which serves us with the requested data is called the server.

4.2 Client-Server Architecture

The client-server architecture is basically a computing model which is hosted by the server to manage the resources and services and be able to deliver the relevant content from the database server when requested by the client [8]. Usually, the client-server architecture function in various tiers –

i. Client and server both operated from the cloud
ii. Client can be local, while server responds from cloud
iii. Database server can be local, while client operates from could.

The “Smart Recommendation System for Music” developed for this project will be for a local client, but database accessed from the cloud (which is Firestore database in this case). For visualizing the client-server architecture, the student has created figure 7.

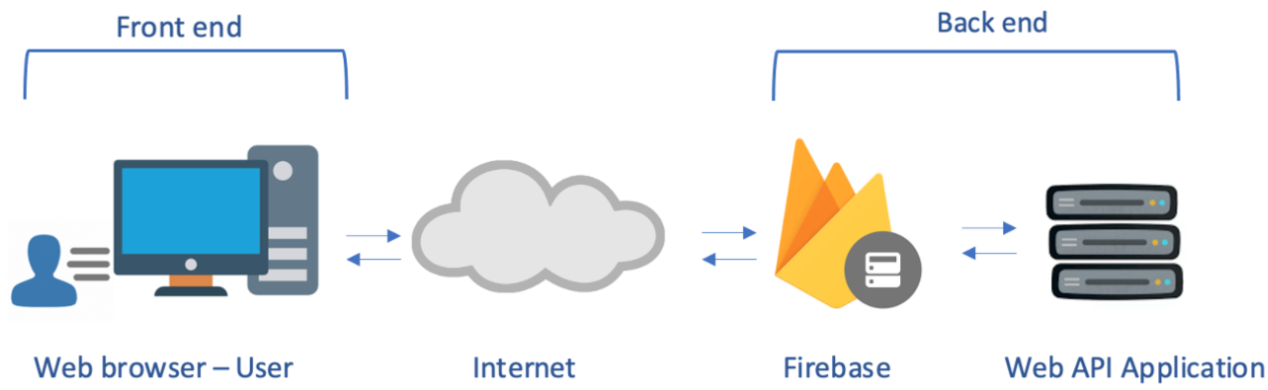


Figure 7: Client structure visualization

Due to various security protocols implemented by different operating systems and browser providers, the client server architecture limits the active data content. Even if we wanted to use Deep Learning Neural Networks, it would be practically impossible to do with an active content requirement for the same reason as security limitations i.e., to exchange dynamic data between the server (the cloud) and the client. With the fact that even the NPM library limits the use of Deep Learning Neural Networks, an approach to implement data analytics with machine learning algorithms based on distance classifiers (which are not library based) such as (1) Euclidean distance (2) Manhattan distance and (3) RBF (4) Nearest neighbours and (5) Cosine Similarity classifiers were discussed with the supervisor and considered the first 3 mentioned.

4.3 GenerateData.com

We have used a website called *generatedata.com* to generate 50 random music albums with 8 features assigned to each album such as (1) Album name (2) Artist name (3) Album year (4) Album genre (5) Song duration (6) Album ranking (7) Album location and (8) Album emotion. Figure 8 below shows the JSON file generated from the website “GenerateData.com”


50GenerateData.json


```json
[
  {
    "Album name": "True",
    "Artist name": "Avicii",
    "Year": 2013,
    "Genre": "Rock",
    "Location": "30.191438848, 43.5211344896",
    "Ranking": 5,
    "Duration": "06:46",
    "Emotion": "Angry"
  },
  {
    "Album name": "Special",
    "Artist name": "Lizzo",
    "Year": 2022,
    "Genre": "R&B",
    "Location": "-49.1823780864, 78.7822343168",
    "Ranking": 21,
    "Duration": "06:34",
    "Emotion": "Happy"
  },
  {
    "Album name": "Love is not dying",
    "Artist name": "Jeremy Zucker",
    "Year": 2020,
    "Genre": "R&B",
    "Location": "-57.5640534016, -61.7916258304",
    "Ranking": 7,
    "Duration": "01:42",
    "Emotion": "Sad"
  },
```


Figure 8: JSON file generated through generatedata.com website

4.4 Firestore/Firebase

A platform developed by Google to enable developers to create web applications or mobile applications such as IOS, android. It has many favourable tools to provide analytics tracking, reporting, and fixing app crashes, and to create marketing and product experiments [4]. In this project we have used firebase to develop our web application with cloud database but local client. We have created a new firestore database called *“Music Recommendation Playground”* and populated all the 50 album collections generated by GenerateData.com into the Firebase using the index.js layout coded with JavaScript. Figure 9 and figure 10 below shows the data of all 50 albums stored in the firebase database using layout 1 from 4.6.

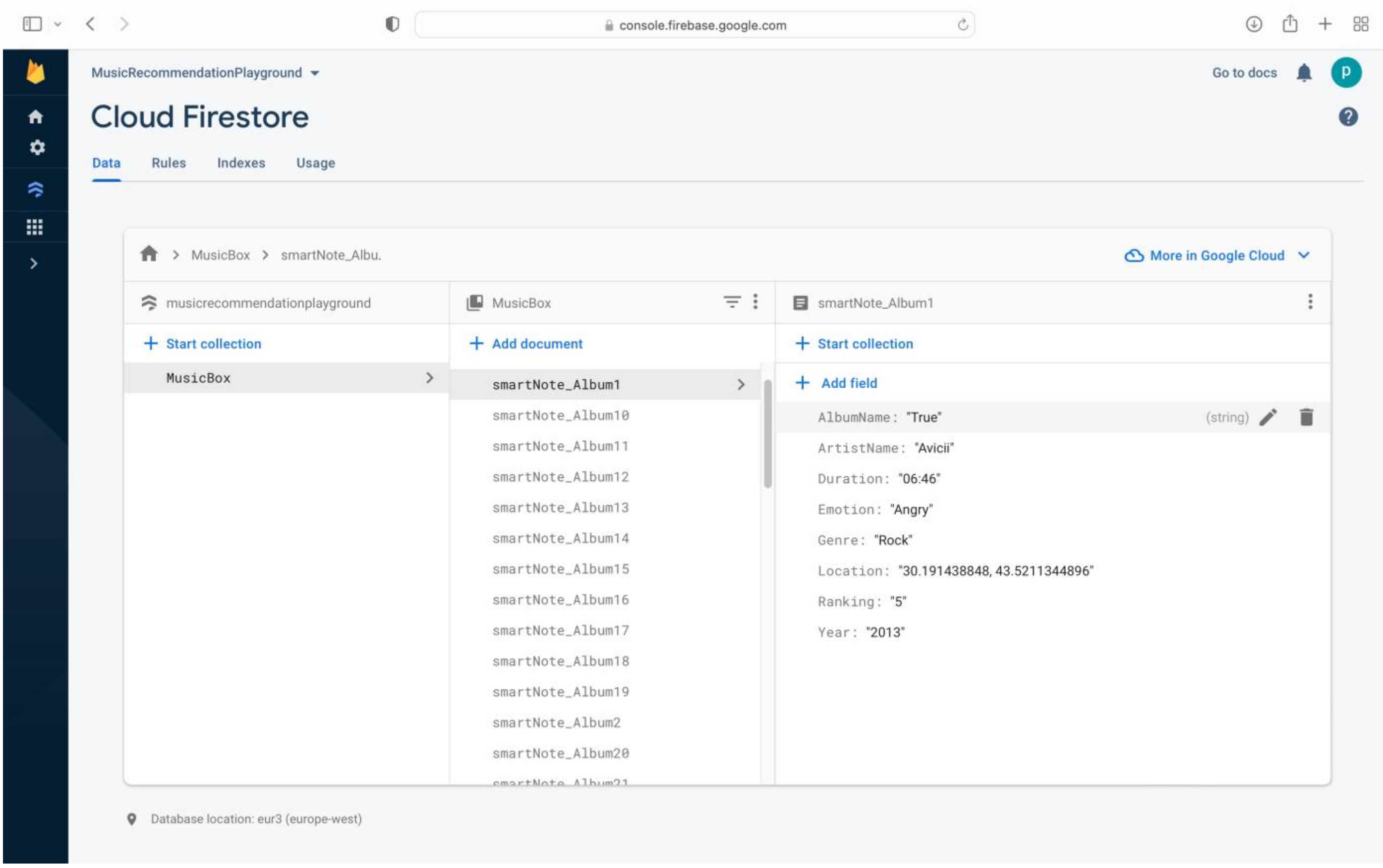


Figure 9: Firestore database of all 50 albums

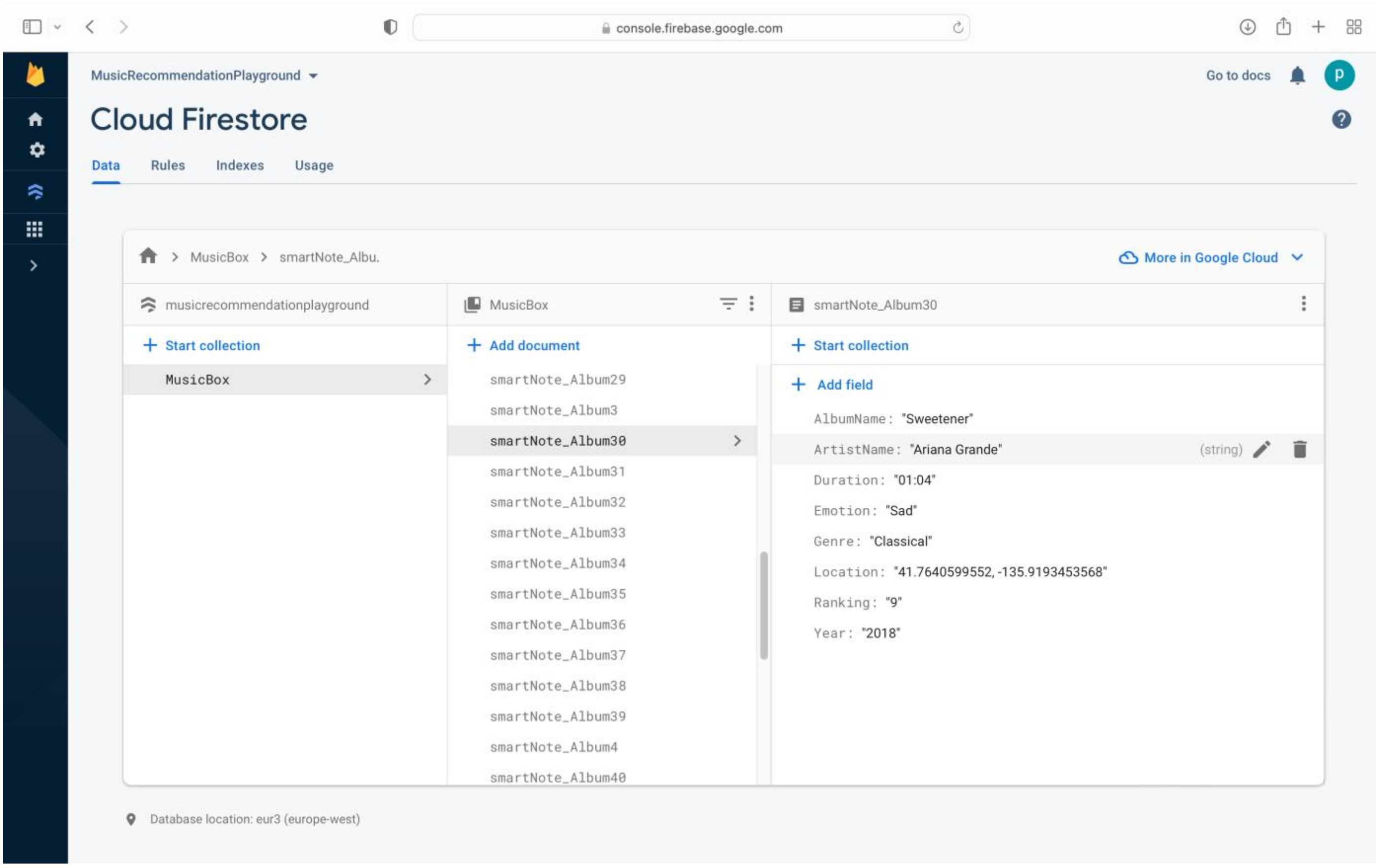


Figure 10: Firestore database of all 50 albums

4.5 Atom and Javascript

Atom is a text editor developed by Github using web technologies. It is an open-source text and source code editor for macOS, Linux and Microsoft windows with plug-ins support to script in JavaScript, html, etc. [5]. JavaScript (JS) is a core programming high-level language. It is most popularly used by the websites on the client's side for webpage behaviours [6]. We have scripted our project's code in Atom IDE using JavaScript and runtime environments such as *node.js, NPM and canvas.js* to run our JSON files outside the browser.

4.6 Web Application Layout

With the project's focus being on the data analytics and implementation of distant classifiers, the web app layout for user interface has been designed with basic concept.

4.6.1 Web App Layout 1

To upload the data of all 50 albums into the firestore database at once, a web app layout has been scripted through a html file. The eight fields for each album as mentioned in 4.3 have been loaded to the firestore database by entering the content in the relevant text fields provided as shown in figure 11.

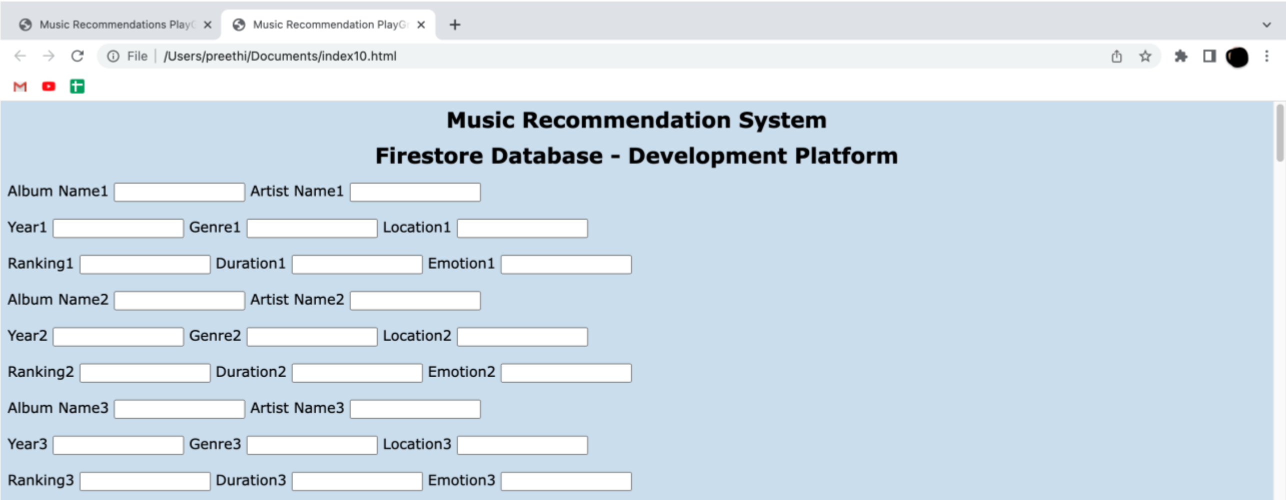


Figure 11: Web App layout to load data into firebase.

After every field of the 50 albums is entered, we can save the entered data on the page by clicking the *"Save"* button provided as shown in figure 12. "Load to firestore" with a "*Load*" button loads the entered data to the firestore database. Layout 1 is for loading the data to Firesore/Firebase purpose only.

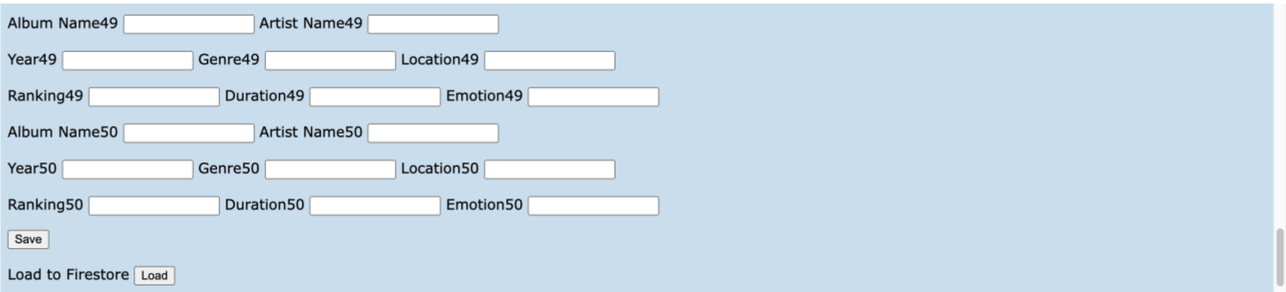


Figure 12: Web App layout to load data into firebase.

4.6.2 *Web App Layout 2*

This is the main layout, with 4 sections as shown in figure 13 and each section is explained below.

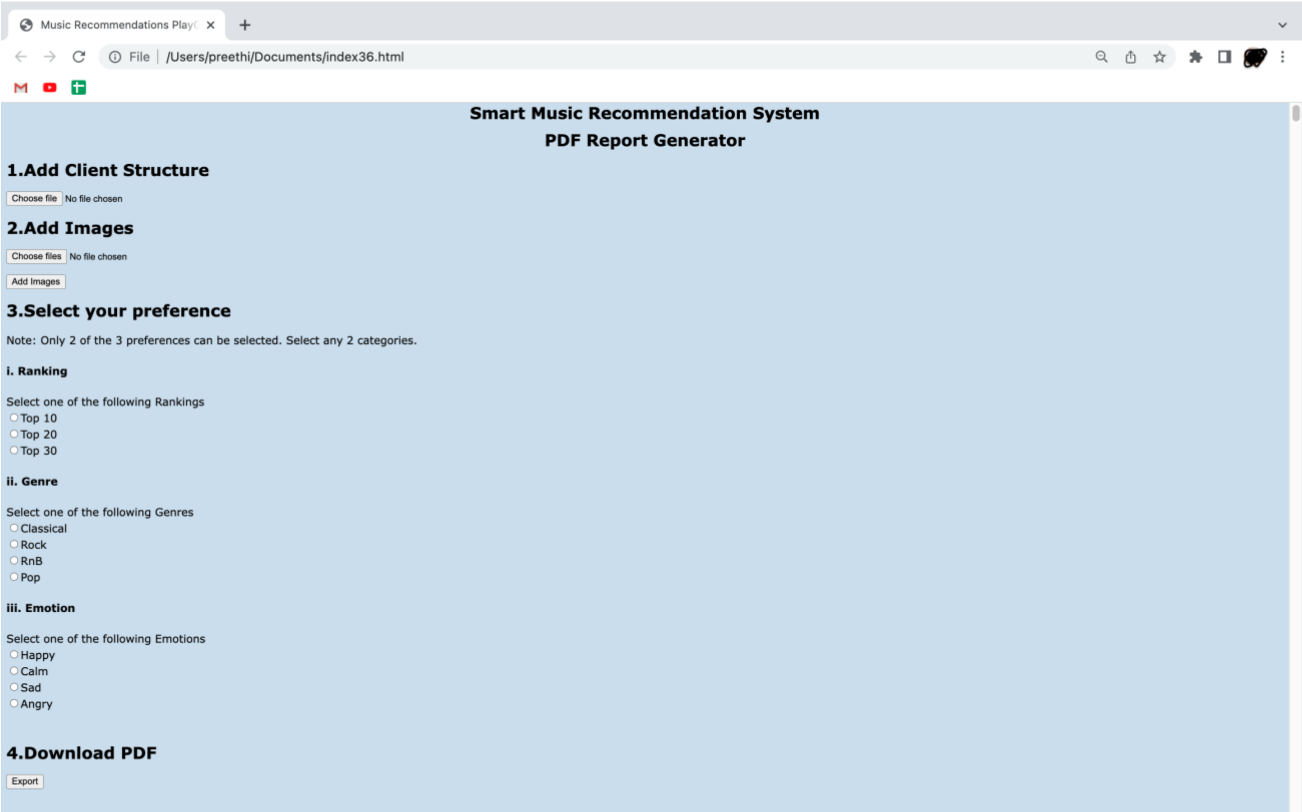


Figure 13: Main Web App layout (GUI)

I. ADD CLIENT STRUCTURE

Here, in the "*choose file*" button, we upload the Music box Client Structure text file defining the size of the PDF table to be printed in the report, containing the list of all 50 albums in the database. An alert on page to confirm the file, pops up once the text file is selected.

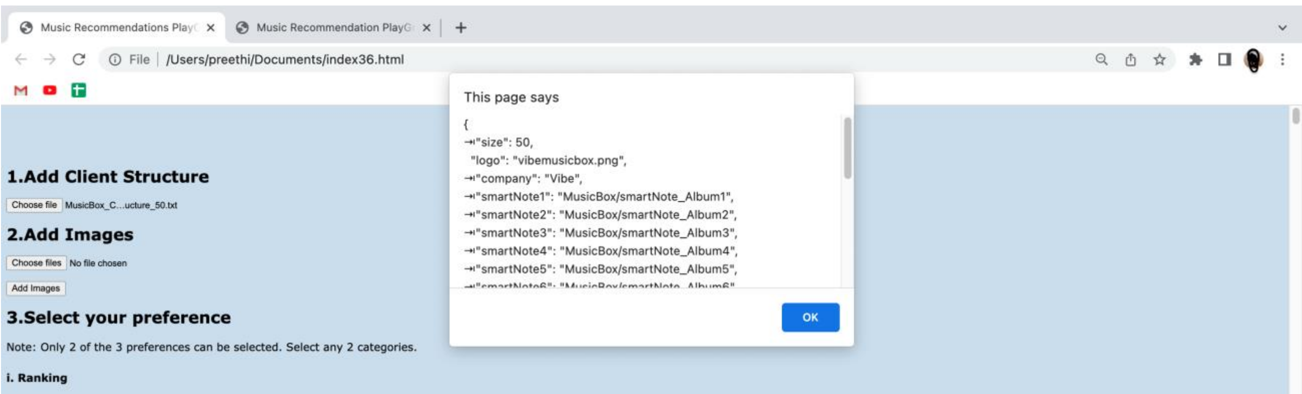


Figure 14: Adding the MusicBox Client Structure

II. ADD IMAGES

Here, in the "*choose file*" button, we. Upload the album cover images for all 50 albums and the logo image for the Smart Music Recommendation System, named in series for the pdf report generator. The 51 chosen files get added to the PDF and the logo is displayed on the web layout when the “*Add Images*” button is clicked. For entertainment purposes, the student has added a logo to the MusicBox and named it “VIBE”.

4.7 Distance classifiers deployed

The focus for this project lies on the backend of the data analytics rather than on the front end of the Graphical User Interface i.e., user functionality is why the web app layout was kept simple and was scripted to perform only the basic conditions needed. A proper understanding of the concept of distance classifiers and its implementation is the indented learning outcome of this project.

To generate the top music recommendations to the user, in this report we deployed analytics algorithms with machine learning i.e., Distance classifiers for a *content-based filtering system* such as –

4.7.1 Euclidean Distance

This technique exposes the shortest distance between two points. Majority of machine learning algorithms along with K-means, use the Euclidean distance metric to scale the similarity between the data. The generalized formula for finding the Euclidean Distance in n-dimensional spaces is [7] (Analytics Vidhya, 2020):

Equation 1: Euclidean Distance →

$$D_e = \left(\sum_{i=1}^{n} (p_i - q_i)^2 \right)^{1/2}$$

The variables:
D = Distance calculated
n = number of dimensions
pi = (x0, y0) data points of user preference in our case
qi = (x1, y1) data points of each album in the database in our case

In simple words, Euclidean distance finds the shortest distance possible between two given points. It is a straight line as demonstrated in figure 15.

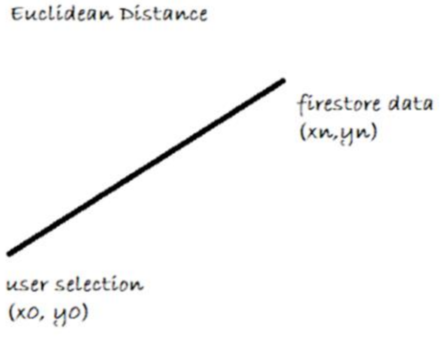


Figure 15: Euclidean straight-line visualization

4.7.2 Manhattan Distance

This machine learning technique gives the total of absolute differences between points among all the dimensions. The generalized formula for finding the Manhattan Distance in n-dimensional spaces is [7] (Analytics Vidhya, 2020):

Equation 2: Manhattan Distance →

$$D_m = \sum_{i=1}^{n} |p_i - q_i|$$

The variables:
D = Distance calculated
n = number of dimensions
pi = (x0, y0) data points of user preference in our case
qi = (x1, y1) data points of each album in the database in our case

In simple words, Manhattan distance is the sum of all possible distance between two given points. It is visualized as a grid layout demonstrated as in figure 16.

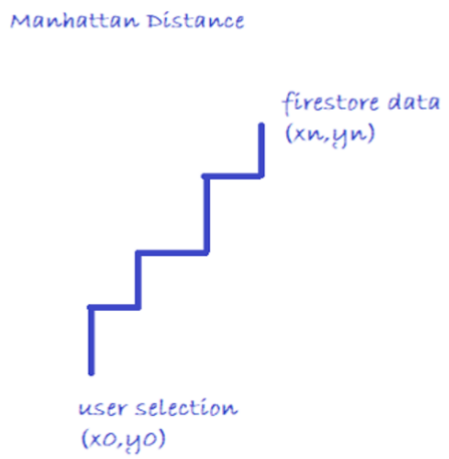


Figure 16: Manhattan Grid layout visualization

4.7.3 Radial Basis Function

This technique was developed to address the issue of choosing the right kernel or feature transform to figure out when dealing with machine learning algorithms. RBF kernels are widely used due to their closeness to Gaussian Distribution. They are also the most generalized version of kernelization [13] (Sreenivasa, 2020). The variance between the two points is what represents the closeness of two pints. They compute the similarity or closeness between two points, which is mathematically represented as:

Equation 3: RBF →

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2})$$

The variables:
K (X1, X2) = Distance calculated
Sigma = The covariance and our hyperparameter
exp = exponential
X1 = (x0, y0) data points of user preference in our case
X2 = (x1, y1) data points of each album in the database in our case
||X1 – X2|| is the Euclidean distance between the two points.

In simple words, radial basis is a variable whose value is determined by the distance between the origin and the mentioned point, in our case the firebase data albums. It is visualized as a bell curved and is similar to Gaussian Kernel. It is visualized as a grid layout demonstrated as in figure 17.

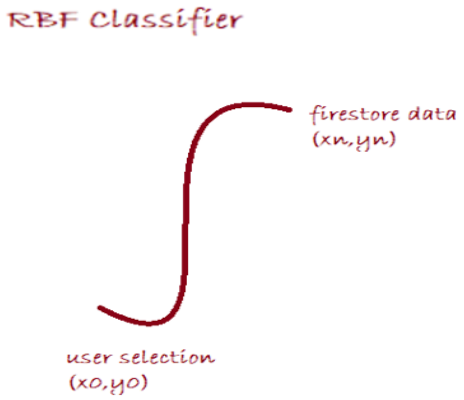


Figure 17: RBF bell curve visualization

The performance of all three distance classifiers is explained below with a diagram. From figure 18, The black dots are referred to the user's preferential selection. The green dots are referred as the similar album matches to the user selection in the database and the red dots are the non-similar matches. The Euclidean distance draws a straight line finding the shortest path, Manhattan distance finds the distance which is the average of all possible distances and Radial Basis Function is supposed to fetch in and find the closest best match, the best curve of the RBF enhances the recommendation by reaching out to the data where Euclidean distance and Manhattan distance could not reach to.

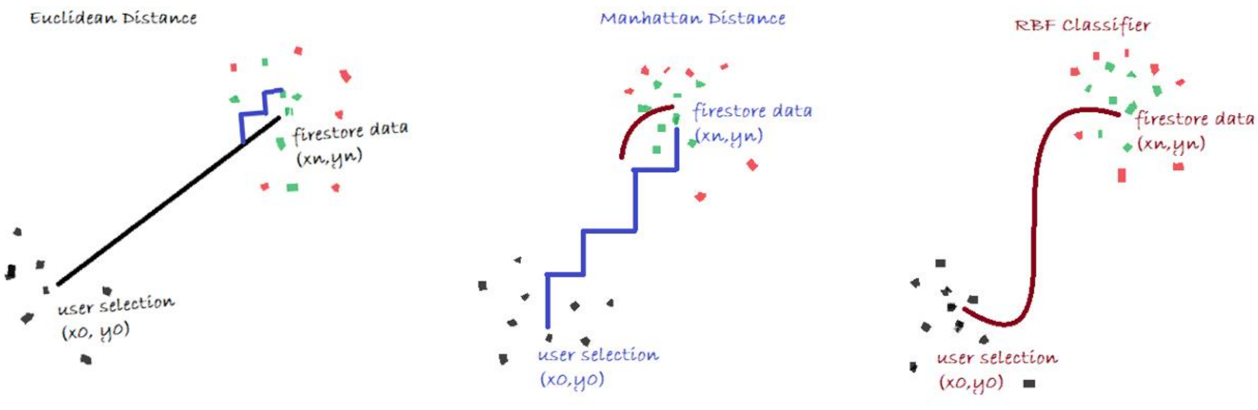


Figure 18: Performance of all 3 distance classifiers

Chapter 5

Results

This section is explained in the following order –

Table 3: Test cases

Case no.	Description	Achieved by
Test Case 1	Design Functionality	Graphical User Interface (GUI)
Test Case 2	User data on music albums preferences	Firestore Data tables processed by the web app of the music recommendation system
Test Case 3	Smart recommendation output – PDF report	PDF of smart recommendation report
Test Case 4	Euclidean Distance classifier output	code snippet from the main.js showing the implementation
Test Case 5	Manhattan Distance classifier output	code snippet from the main.js showing the implementation
Test Case 6	RBF Distance classifier output	code snippet from the main.js showing the implementation

5.1 Test Case 1 - Design functionality

As briefly explained with figures 11 – 14, in chapter 4 - layout 2, the web application layout (which is the Graphical User Interface) is designed with simplicity as our learnings were more on implementing data analytics. The GUI web layout has functioned as scripted. The console tab shows all the code executed when relevant buttons are clicked, see figure 19.

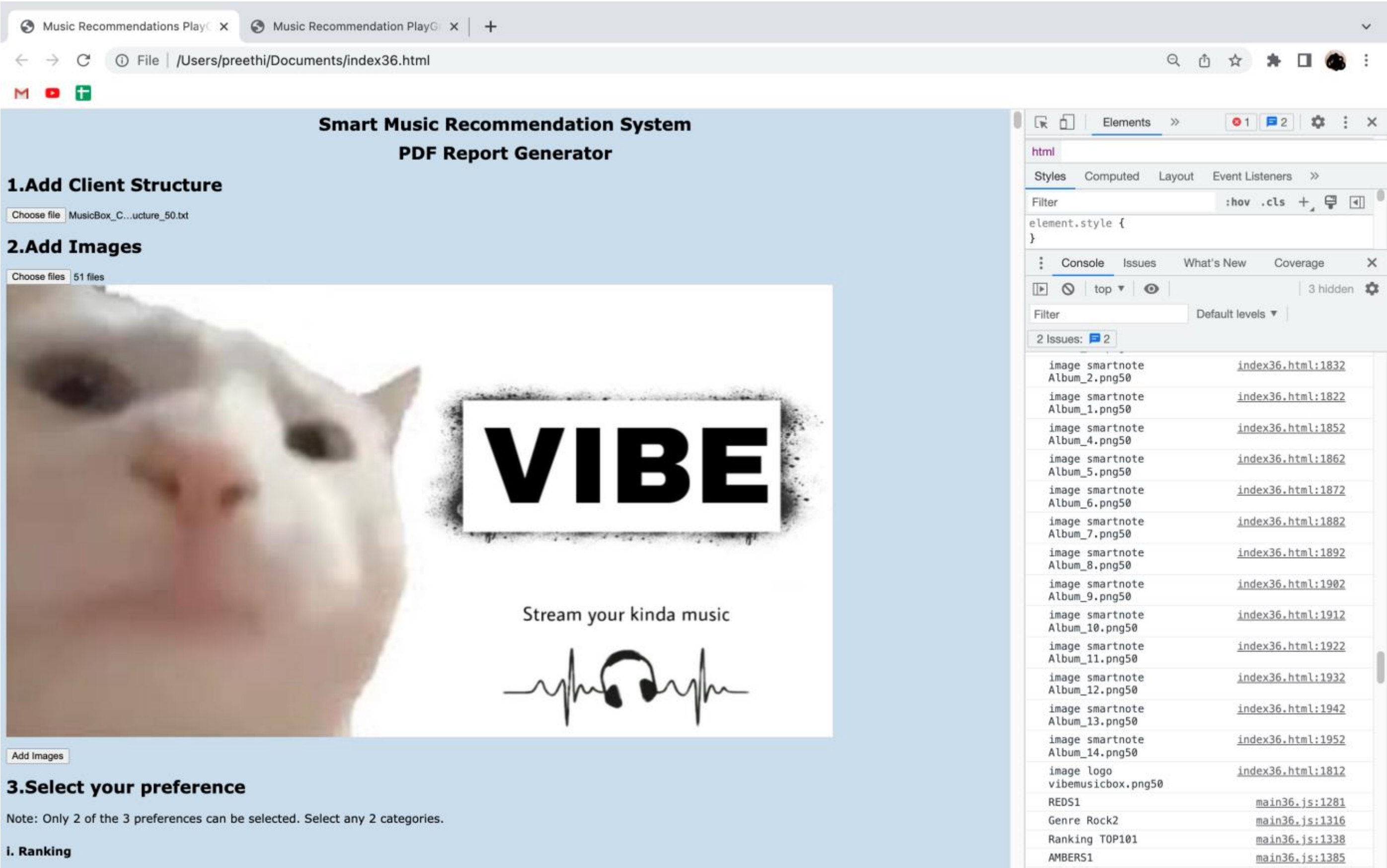


Figure 19: Web out layout functioning

The radio buttons are disabled as scripted that when any two selections are made from the three categories mentioned i.e., the three radio group buttons. Once the "*Export*" button is clicked, the closeness of each album in the database is calculated in three different methods and are pushed into an arrow. This array is later sorted based on the score of each album in ascending order. Then the sorted array is sliced into 3 to get the top 3 suggestions displayed in the table. From the figure 20, the console log displays the comments when the code is executed and when recommendations are made by the 3 distant classifiers.

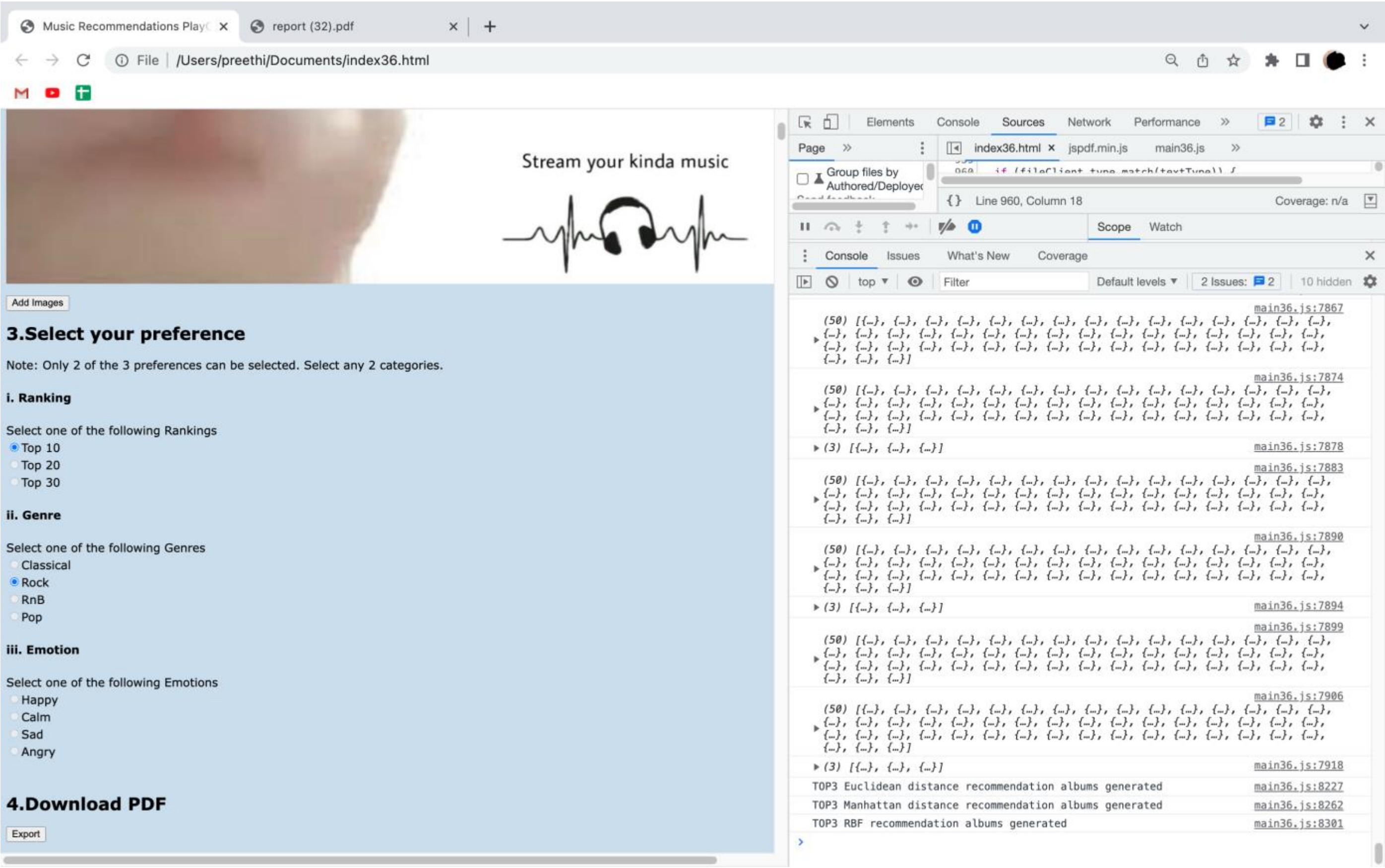


Figure 20: Web app layout functioning

5.2 Test Case 2 – User Data on Music Albums' Preferences

Figure 21 shows the data of all 50 albums stored in the firebase database using layout 1 from 4.6. A collection called "MusicBox" is the data file for our music recommendation system. This collection stores the record of all 50 documents defined as "Albums" uploaded through the web app layout 1. Each album with 8 fields i.e., 8 subcategories of data: album name, artist name, duration of the songs, emotion of the album, genre of the album, location of the album origin, ranking of the album songs, and year it was released.

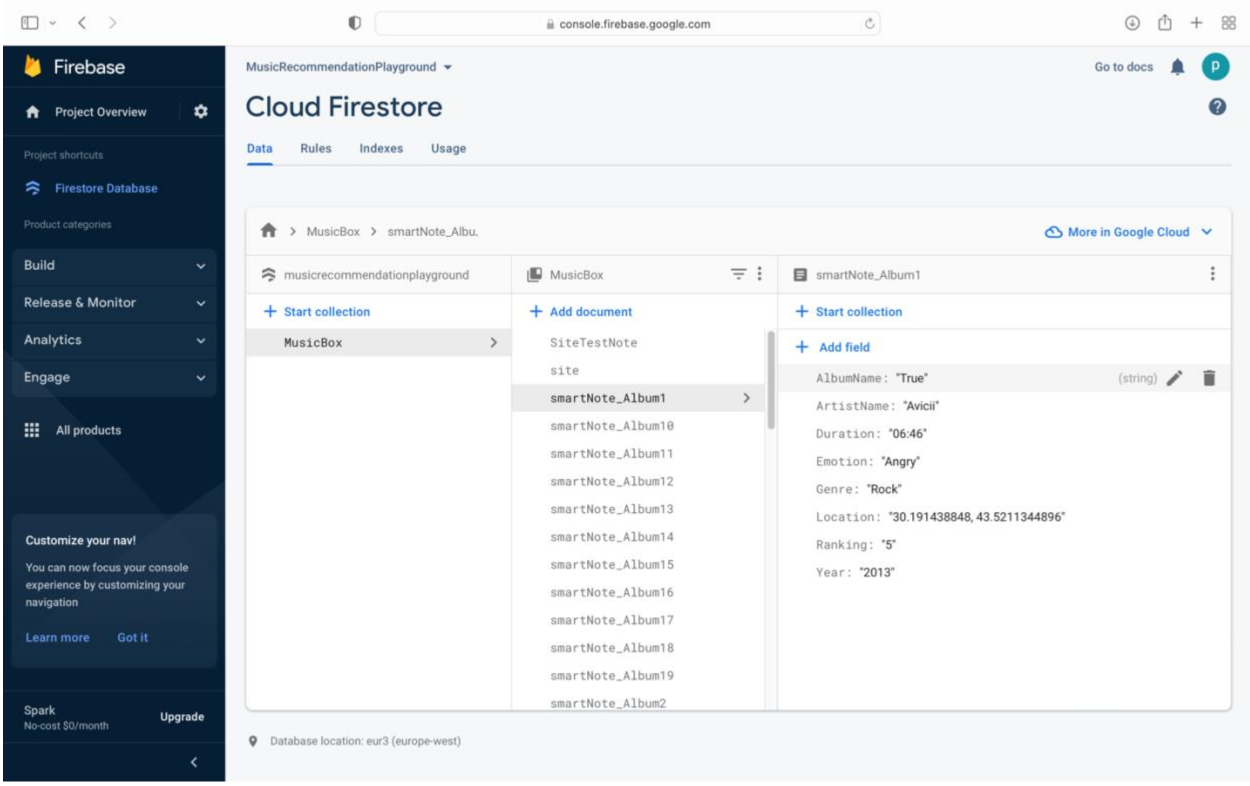
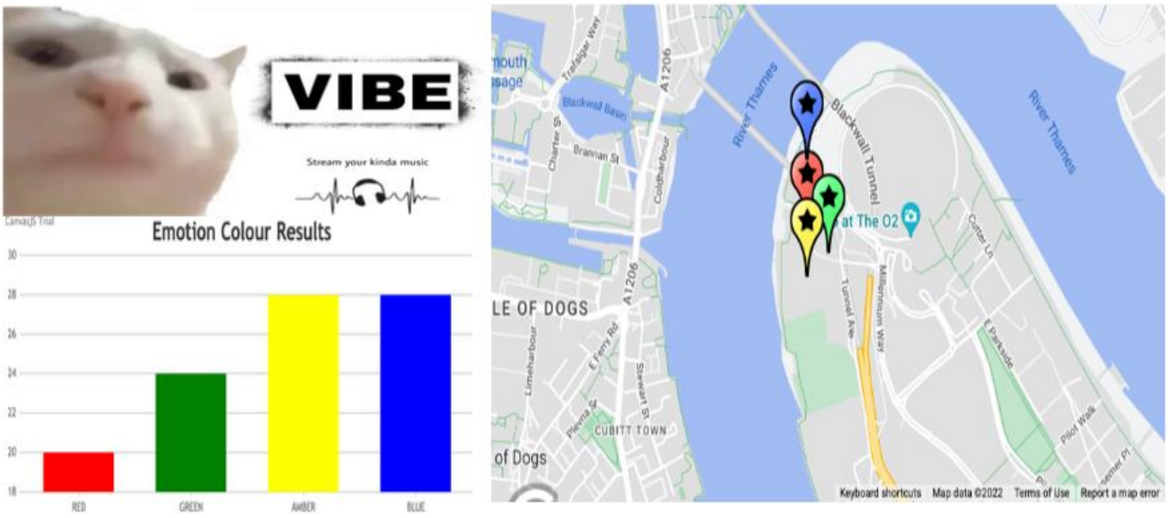


Figure 21: Firestore Database of 50 albums uploaded through layout 1

5.3 Test Case 3 – Smart Recommendation Output – PDF Report



Album Name	Year	Artist Name	Genre	Emotion	Album Cover
True	2013	Avicii	Rock	Angry	
Special	2022	Lizzo	R&B	Happy	
Love Is Not Dying	2020	Jeremy Zucker	R&B	Sad	
Youth	2016	BTS	Pop	Calm	
Vice Versa	2021	Rauw Alejandro	Pop	Happy	
A Girl Like Me	2006	Rihanna	Classical	Angry	
Believe	2011	Justin Bieber	Pop	Happy	
Lemonade	2011	Beyonce	R&B	Calm	
Evermore	2010	Taylor Swift	R&B	Calm	
The Dutchess	2006	Fergie	Classical	Angry	
Love Goes	2020	Sam Smith	Classical	Sad	

Figure 22: First page of the PDF generated

When the "*Export"* button is clicked, a 5 page - PDF is generated. The first page having 3 sections on the top followed by a table displaying the details of all the available albums (50 albums) from the firestore database.

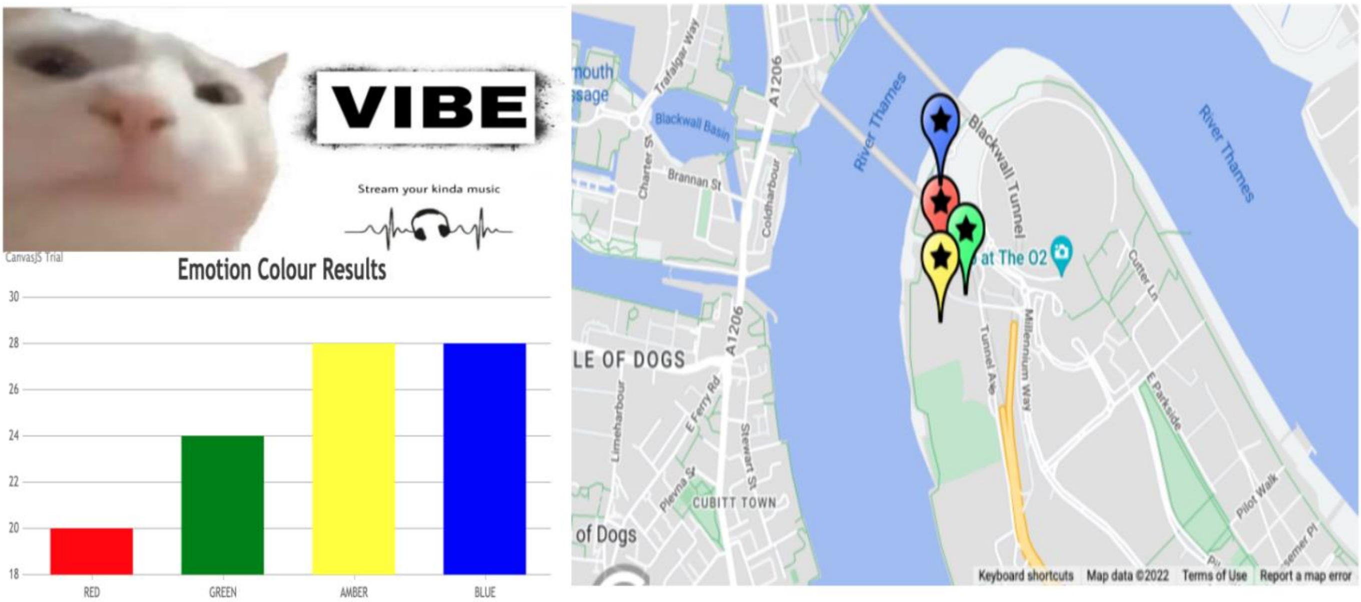


Figure 23: Three informative sections on the top of PDF

From figure 23, the top left image is the logo of the Music Box called "VIBE". The bottom left image is a bar graph displaying the frequency of 4 emotional categories across the database. Basic data analytics were calculated for the "emotion" category. The map on the right displays the location of albums with points pinned to the geolocation defined. Attempt to use a customised map was made, but the website restricted the service due to billing details not being provided. Due to this issue, th supervisor has suggested to use a standard map provided by the supervisor.

```
//=======================================================================//
  //DATA ANALYTICS ALGORITHMS HERE
  // calculating the 4 categories of emotion
  var numEmotionsCats = (redsCounter+greensCounter+ambersCounter+bluesCounter);
  var percentageGreen = (greensCounter*100)/numEmotionsCats;
  var percentageRed = (redsCounter*100)/numEmotionsCats;
  var percentageAmber = (ambersCounter*100)/numEmotionsCats;
  var percentageBlue = (bluesCounter*100)/numEmotionsCats;
  var emotionCatsResults = [percentageGreen, percentageRed, percentageAmber, percentageBlue];
  var green = emotionCatsResults[0];
  var red = emotionCatsResults[1];
  var amber = emotionCatsResults[2];
  var blue = emotionCatsResults[3];
  //----------------------------------------------------------------------//
```
Figure 24: Code snippets of data analytics for classifying four Emotion categories

The last page displayed top 3 recommendations in each table, from each distance classifier based on user preferences selected on the layout. Results discussed in relevant sections below. More images attached to the appendix.

The distance classifiers have been implemented with three conditions i.e., for 2 out of 3 choices of possible selection made –

I. Ranking and Genre
II. Genre and Emotion
III. Ranking and Emotion

With the user selection of preferences set to “Top 10” and “Rock” as shown in the figure 25, the PDF generated 3 tables with top 3 suggestions from the three distance classifiers implemented, each discussed in relevant sections below.

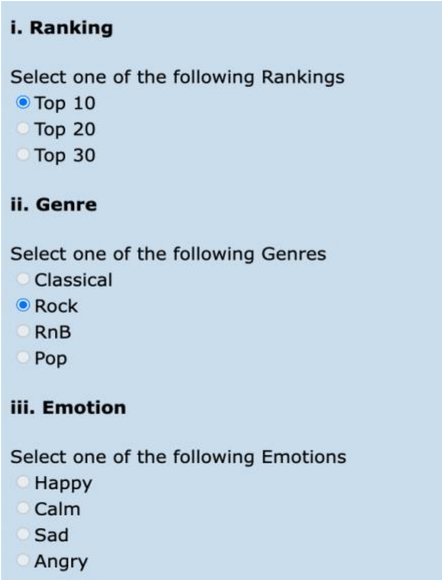


Figure 25: Radio buttons selected for testing

5.4 Test Case 4 – Euclidean Distance Classifier Output

As explained earlier in 4.7, the Euclidean distance was formulated as DistED as shown in equation 4. Where, points (x0, y0) will be the two user selection variables and points (x1, y1) will be the respective fields of all 50 albums in the firestore firebase.

Equation 4 → **DistED = SquareRoot((x0-x1)^2+(y0-y1)^2)**

```
//FUNCTION FOR EUCLIDEAN DISTANCE
function getEuclideanDistance(x0_emotion, y0_ranking, z0_genre, xn_emotionFn, yn_rankingFn, zn_genreFn)
{
    var distED = 0;

    if (z0_genre == null)
    {
      //Calculating the distance between two points
      distED = Math.sqrt( Math.pow((x0_emotion-xn_emotionFn), 2) + Math.pow((y0_ranking-yn_rankingFn), 2) );
      console.log("Euclidean distance is: " + distED);
      return distED;
    }
    else if (x0_emotion == null)
    {
      //Calculating the distance between two points
      distED = Math.sqrt( Math.pow((z0_genre-zn_genreFn), 2) + Math.pow((y0_ranking-yn_rankingFn), 2) );
      console.log("Euclidean distance is: " + distED);
      return distED;
    }
    else if (y0_ranking == null)
    {
      //Calculating the distance between two points
      distED = Math.sqrt( Math.pow((z0_genre-zn_genreFn), 2) + Math.pow((x0_emotion-xn_emotionFn), 2) );
      console.log("Euclidean distance is: " + distED);
      return distED;
    }
} // end function getEuclideanDistance
```

Figure 26: Code snippets of Euclidean Distance Function

The last page of the PDF generated contains 3 tables. The first one with top three recommendations generated by the Euclidean Distance classifier for the user selection of “Top 10” in ranking and “Rock” in Genre. With the database having only 2 albums in that combination, the distance classifier still suggested an album closer to the user selection, as you can see in the 3rd suggestion on the table.

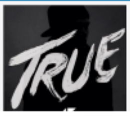
Euclidean Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
1	True	Avicii	5	Rock	Angry	
1	24K Magic	Bruno Mars	6	Rock	Happy	
1	Whitney	Whitney Houston	12	Rock	Calm	

Figure 27: Top 3 recommendations by Euclidean Distance

5.5 Test Case 5 – Manhattan Distance Classifier Output

As explained earlier in 4.7, the Manhattan distance was formulated as DistMD as shown in equation 5. Where, points (x0, y0) will be the two user selection variables and points (x1, y1) will be the respective fields of all 50 albums in the firestore firebase.

Equation 5 → **DistMD = ||(x1-x0) + (y1-y0)||**

```
//FUNCTION FOR MANHATTAN DISTANCE
function getManhattanDistance(x0_emotion, y0_ranking, z0_genre, xn_emotionFn, yn_rankingFn, zn_genreFn)
{
    var distMD = 0;

    if (z0_genre == null)
    {
      //Calculating the distance between two points
      distMD = Math.abs(xn_emotionFn - x0_emotion) + Math.abs(yn_rankingFn - y0_ranking);
      console.log("Manhattan distance is: " + distMD);
      return distMD;
    }
    else if (x0_emotion == null)
    {
      distMD = Math.abs(zn_genreFn - z0_genre) + Math.abs(yn_rankingFn - y0_ranking);
      console.log("Manhattan distance is: " + distMD);
      return distMD;
    }
    else if (y0_ranking == null)
    {
      distMD = Math.abs(zn_genreFn - z0_genre) + Math.abs(xn_emotionFn - x0_emotion);
      console.log("Manhattan distance is: " + distMD);
      return distMD;
    }
} // end function getManhattanDistance
```

Figure 28: Code snippets of Manhattan Distance Function

The last page of the PDF generated contains 3 tables. The second table contains top three recommendations generated by the Manhattan Distance classifier for the user selection of “Top 10” in ranking and “Rock” in Genre. With the database having only 2 albums in that combination, the distance classifier still suggested an album closer to the user selection, as you can see in the 3rd suggestion on the table. Observation here is that Manhattan Distance tends to give suggestions like Euclidean distance.

Manhattan Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
0	True	Avicii	5	Rock	Angry	
0	24K Magic	Bruno Mars	6	Rock	Happy	
0	Whitney	Whitney Houston	12	Rock	Calm	

Figure 29: Top 3 recommendations by Manhattan Distance

5.6 Test Case 6 – RBF Distance Classifier Output

First, we find both the means of the two variables the user has selected. MeanX and meanY of the user selected preferences value and each album respective fields value in the database, to calculate the coVariance. Where, points (x0, y0) will be the two user selection variables and points (x1, y1) will be the respective fields of all 50 albums in the firestore firebase. The mean was formulated as [16] –

Equation 6 → **meanX = (x0 + x1)/2**
Equation 7 → **meanY = (y0 + y1)/2**

```
// Function to find mean.
function findmean(x0_emotion, y0_ranking, z0_genre, xn_emotion, yn_ranking, zn_genre)
{
  var meanX = 0;
  var meanY = 0;

  if (z0_genre == null) // if genre is not selected
  {
    meanX = ((parseInt(x0_emotion)) + xn_emotion)/2;
    meanY = ((parseInt(y0_ranking)) + yn_ranking)/2;

    return [meanX, meanY];
  }
  else if (y0_ranking == null)  // if ranking is not selected
  {
    meanX = ((parseInt(x0_emotion)) + xn_emotion)/2;
    meanY =  ((parseInt(z0_genre)) + zn_genre)/2;

    return [meanX, meanY];
  }
  else if (x0_emotion == null)  // if emotion is not selected
  {
    meanX = ((parseInt(z0_genre)) + zn_genre)/2;
    meanY = ((parseInt(y0_ranking)) + yn_ranking)/2;

    return [meanX, meanY];
  }
} // end findmean function
```

Figure 30: Code snippets of finding Mean to find coVariance

After finding the meanX and meanY, we calculate the coVariance for each album in the database to one of the 3 conditions based on what the user has selected. Covariance is formulated as shown below [16] -

Equation 8 → **coVariance = ((x0-meanX) (y0-meanY) + (x1-meanX) (y1-meanY))/2**

```
// function to find coVariance
function findcoVariance(x0_emotion, y0_ranking, z0_genre, xn_emotion, yn_ranking, zn_genre, meanX, meanY)
{
    var coVariance = 0;

    if (z0_genre == null)
    {
      coVariance = (( (((parseInt(y0_ranking))-meanY) * ((parseInt(x0_emotion)) - meanX)) + ((yn_ranking-meanY)*(xn_emotion-meanX)) )/2);
    }
    else if (y0_ranking == null)
    {
      coVariance = (( (((parseInt(x0_emotion))-meanX) * ((parseInt(z0_genre)) - meanY)) + ((xn_emotion-meanX)*(zn_genre-meanY)) )/2);
    }
    else if (x0_emotion == null)
    {
      coVariance = (( (((parseInt(z0_genre))-meanX) * ((parseInt(y0_ranking)) - meanY)) + ((zn_genre-meanX)*(yn_ranking-meanY)) )/2);
    }
    if (coVariance == 1)
    {
      return coVariance = 0.99;
    }
    else if (coVariance == 0)
    {
      return coVariance = 0.01;
    }
    else if (0<coVariance && coVariance<1)
    {
      return coVariance;
    }
} // end findcoVariance function
```

Figure 31: Code snippets of finding covariance for Radial Basis Function

The Radial Basis Function (RBF) was formulated as DistRBF according to [13] (Sreenivasa, 2020) -

Equation 9 → **DistRBF = exp(-((x1-x2)^2+(y1-y2)^2))/2*(coVariance^2))**

```
//FUNCTION FOR RBF
function getRBF(x0_emotion, y0_ranking, z0_genre, xn_emotionFn, yn_rankingFn, zn_genreFn, coVariance)
{
    var distRBF = 0;

    if (z0_genre == null)
    {
      //Calculating the distance between two points
      distRBF = Math.exp(- (Math.pow((x0_emotion - xn_emotionFn),2) - Math.pow((y0_ranking - yn_rankingFn), 2))/(2*(Math.pow(coVariance),2)));

      console.log("RBF value is: " + distRBF);
      return distRBF;
    }
    else if (x0_emotion == null)
    {
      distRBF = Math.exp(- (Math.pow((z0_genre - zn_genreFn),2) - Math.pow((y0_ranking - yn_rankingFn), 2))/(2*(Math.pow(coVariance),2)));
      console.log("RBF value is: " + distRBF);
      return distRBF;
    }
    else if (y0_ranking == null)
    {
      distRBF = Math.exp(- (Math.pow((z0_genre - zn_genreFn),2) - Math.pow((x0_emotion - xn_emotionFn), 2))/(2*(Math.pow(coVariance),2)));
      console.log("RBF value is: " + distRBF);
      return distRBF;
    }
} // finction end getRBF
```

Figure 32: Code snippets of Radial Basis Function

The last page of the PDF generated contains 3 tables. The second table contains top three recommendations generated by the Radial Basis Function Distance classifier for the user selection of “Top 10” in ranking and “Rock” in Genre. With the database having only 2 albums in that combination, the distance classifier still suggested an album closer to the user selection, as you can see in the 3rd suggestion on the table.

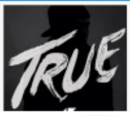
RBF Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
0.778800783071404	True	Avicii	5	Rock	Angry	
0.778800783071404	24K Magic	Bruno Mars	6	Rock	Happy	
0.778800783071404	Whitney	Whitney Houston	12	Rock	Calm	

Figure 33: Top 3 recommendations by Radial Basis Function

The co-variance managed to work accurately for certain user selection combinations but has been generating the same results as to Euclidean distance and Manhattan distance. Due to this reason, the student has eliminated the calculation of the co-variance for each album and assumed it to be the value of 1 by default. According to the article [13] (Sreenivasa, 2020), the RBF can be calculated with Covariance set to 1, in that case, the points with the highest distance will be the points that are classes to the origin. In this case the formula for calculating the RBF distance would become equation 10:

Equation 10: RBF formula when covariance = 1 →

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2})$$

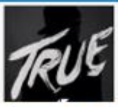
The JavaScript files names for the code are submitted in 2 sets

Set 1: Covariance calculated for each album in the database
- Index_MRS.html
- Main_MRS.js
- MyMap_MRS.js

Set 2: Covariance assumed to be 1 by default for all albums (without covariance calculation)
- Index_MRS_woCov.html
- Main_MRS_woCov.js
- MyMap_MRS_woCov.js

5.7 Example of a difficult test case

A ‘difficult’ test case was created and performed on, to test the performance of RBF effectively. A user’s preferential combination of inputs which does not exist in the albums form the database was given as inputs to see how the distance classifiers would perform. An album with a combination of *“Rock” genre* and *“Calm” emotion* were given as inputs. The test case was performed on Set 2 code i.e., where covariance is calculated for each of the album. The generated PDF had the following results.

Euclidean Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
1	True	Avicii	5	Rock	Angry	
1	Melodrama	Lorde	4	Rock	Angry	
1	Anti	Rihanna	25	Rock	Angry	

Manhattan Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
1	True	Avicii	5	Rock	Angry	
1	Melodrama	Lorde	4	Rock	Angry	
1	Anti	Rihanna	25	Rock	Angry	

RBF Score	Album Name	Artist Name	Ranking	Genre	Emotion	Album Cover
2.718282	Lemonade	Beyonce	25	R&B	Calm	
2.718282	Evermore	Taylor Swift	9	R&B	Calm	
2.718282	24K Magic	Bruno Mars	6	Rock	Happy	

Figure 34: Difficult test case results

From figure 34, it can be observed that, when a combination of inputs that does not exist in the database was given to test, Euclidean distance and Manhattan Distance have mad recommendation on songs that satisfy one of the conditions i.e., "*Rock"* genre and nothing of "*calm"* emotion songs. But in case of Radial Basis Function, it can be observed that it has managed to suggest songs which satisfied both the conditions individually i.e., two songs with "*Calm"* emotion and one song with "*Rock"* genre.

From this test case experiment, it can be observed that RBF outperforms Euclidean distance and Manhattan distance in challenging situations.

Chapter 6

Discussion

Euclidean distance, Manhattan distance, and Gaussian form - Radial Basis Function were the three distance classifiers deployed in this project for similarity measurement between user choices and firebase database. Euclidean distance compares the distance between the user preference point and each of the 50 albums and finds the shortest possible path. Manhattan distance is calculated by adding all the possible path distances between the user selection points and each of the 50 albums. While the Radial Basis Function is the squared exponential covariance. From the recommendation generated through the PDF (figures provided in the appendix), we can observe that the Euclidean distance and Manhattan distance are generating like results although it can be said that Manhattan is performing slightly better than Euclidean distance with other combinations of user selection. But in most cases the results of Euclidean distance and Manhattan distance are similar. Radial Basis Function seems to be performing slightly better than Manhattan distance with better results for some combinations of the user selections, however, in most cases it is generating similar results to Euclidean distance and Manhattan distance.

Overall, the objectives of this project were successfully implemented and delivered as planned. The PDF report generated contains the logo, frequency of 4 emotion categories, map with 4 pinpoints of album locations, a pdf table with details of 50 albums in the database, and 3 tables with top 3 recommendations suggested by the 3 distant classifiers. The tests and observations (see appendix for more images) confirm that Radial Basis Function is performing recommendations better than or similar to Manhattan Distance and Manhattan Distance is performing recommendations better than or similar to Euclidean distance.

The Web-Client architecture limited the active content data which restricts us from implementing any deep learning neural networks as we cannot access default libraries even if we wanted to. “Active content is a type of interactive or dynamic website content that includes programs like Internet polls, JavaScript applications, stock tickers, animated images, ActiveX applications, action items, streaming video and audio, weather maps, embedded objects, and much more. Active content contains programs that trigger automatic actions on a Web page without the user's knowledge or consent. Web developers use active content to visually enhance the Web page or provide additional functionality beyond basic HTML” [10]. The calculation for the covariance can be improvised in the future works as this project limits the covariance accuracy for certain combinations of inputs. Limitations on the linear orthogonality assumption, which is, the user is only allowed to select 2 out of 3 options from the radio buttons provided, the radio button were scripted to be disabled after a maximum of 2 selections, so the distant classifiers need to check with only 3 conditions and be executed. The more selections the user is allowed to make, that’s how accurate the suggestions will be.

Chapter 7

Conclusion and future work

For this project, a Smart Music Recommendation System (SMRS) has been developed and presented. Three distance classifiers which are machine learning - similarity measurement techniques were deployed to generate top 3 recommendations on the PDF. A dataset of 50 music album with eight categories: 1) Album Name 2) Artist name 3) Emotion 4) Genre 5) Ranking 6) Location 7) Year 8) Duration, was generated and used. It was stored in the firebase database and accessed through a web layout for graphical User Interface. Two Web app layouts were scripted.

I. To load the 50 albums to the firestore database.
II. Main layout to add the client structure, 50 album cover images and provide inputs for the PDF to be generated i.e., 2 input through 3 radio buttons groups.

The generated resultant PDF contained three images on the top displaying the logo information, 4 emotion categories in the database information and the map with 4 album locations pinned. Then a table with details of all 50 albums in the database with images starts under the 3 images and is followed by the three tables each containing top 3 recommendations from three distance classifier mentioned. Through the main layout, the user uploads the client structure, uploads the image files of 50 albums to be added in the PDF, and enters his preferences through the radio buttons.

From the results and the test cases discussed in Chapter 5, it can be observed that the Radial Basis Function generates the best results in difficult cases compared to Euclidean distance and Manhattan distance. However, in most of the cases the three distance classifiers generate the same results. These are the results when the covariance is assumed to be 1. With covariance calculated for each album individually would have given out best results. This is one improvement that the student would like to suggest to the future projects taken in this area.

By observing the limitations in Chapter 6, the current data analytics used for this project can be improved with some suggestions drawn through observation. To calculate similarity in higher dimensions or for multiple classes (i.e., more than two user selections given), implementation of other machine learning techniques such as (1) K – Nearest Neighbours algorithm, which is based on instant learning and works well with big datasets, (2) cosine similarity algorithm, (3) A*, etc. Techniques for dimensionality reduction can be implemented in this project to enhance the outcomes in case of curse of dimensionality, which brings us to “Manifold Learning” which uses a high dimensional data to produce a low dimensional projection of it [15] (Brownlee, 2020). Furthermore, according to (Kapasi. H, 2020) [16] non-linear dynamic systems can be modelled with neural networks with a NARX, a standard technique I.e., a nonlinear generalization of the autoregressive exogenous.

Acknowledgement

This project was developed for the student’s MSc’s final project with 3 - months duration at De Montfort University. The objectives have been successfully delivered along with the learning outcomes. The student sincerely thanks the supervisor - Dr. Ioannis Kypraios for his support, guidance, and motivation throughout the project.

References

Chapter 2 references

[1]

Deng, S., Wang, D., Li, X. and Xu, G. (2015). Exploring user emotion in microblogs for music recommendation. *Expert Systems with Applications*, 42(23), pp.9284–9293. doi:10.1016/j.eswa.2015.08.029.

[2]

Rosa, R.L., Rodriguez, D.Z. and Bressan, G. (2015). Music recommendation system based on user's sentiments extracted from social networks. *IEEE Transactions on Consumer Electronics*, 61(3), pp.359–367. doi:10.1109/tce.2015.7298296.

[3]

Bodarwé, K.-A., Noack, J. and Jean-Jacques, P. (2015). Emotion-based music recommendation using supervised learning. *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. doi:10.1145/2836041.2841204

[4]

Su, J. and Zhou, P. (2022). Machine Learning-based Modeling and Prediction of the Intrinsic Relationship between Human Emotion and Music. *ACM Transactions on Applied Perception*. doi:10.1145/3534966.

[5]

Singhania, D. (2020). *Music genre analysis — Understanding emotions and topics in different genres*. [online] Medium. Available at: https://towardsdatascience.com/music-genre-analysis-understanding-emotions-and-topics-in-different-genres-110377d8888c

[6]

Boyd, C. (2019). *How Spotify Recommends Your New Favorite Artist*. [online] Medium. Available at: https://towardsdatascience.com/how-spotify-recommends-your-new-favorite-artist-8c1850512af0.

[7]

Dwivedi, P. (2019). *Using CNNs and RNNs for Music Genre Recognition*. [online] Medium. Available at: https://towardsdatascience.com/using-cnns-and-rnns-for-music-genre-recognition-2435fb2ed6af.

[8]

Ting, I-Hsien. and Yu, P.-L. (2016). *Proceedings of The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society*. doi:10.1145/2925995.2926022.

[9]

Knees, P. and Schedl, M. (2013). A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(1), pp.1–21. doi:10.1145/2542205.2542206.

[10]

Wang, R., Ma, X., Jiang, C., Ye, Y. and Zhang, Y. (2020). Heterogeneous information network-based music recommendation system in mobile networks. *Computer Communications*, [online] 150, pp.429–437. doi:10.1016/j.comcom.2019.12.002.

[11]

Chiu, M.-C., Huang, J.-H., Gupta, S. and Akman, G. (2021). Developing a personalized recommendation system in a smart product service system based on unsupervised learning model. *Computers in Industry*, 128, p.103421. doi:10.1016/j.compind.2021.103421.

[12]

Ludewig, M., Kamehkhosh, I., Landia, N. and Jannach, D. (2018). Effective Nearest-Neighbor Music Recommendations. *Proceedings of the ACM Recommender Systems Challenge 2018 on - RecSys Challenge ’18*. doi:10.1145/3267471.3267474.

[13]

Schedl, M. (2013). Ameliorating Music Recommendation. *Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM ’13*. doi:10.1145/2536853.2536856.

[14]

Wang, X. and Wang, Y. (2014). Improving Content-based and Hybrid Music Recommendation using Deep Learning. *Proceedings of the ACM International Conference on Multimedia - MM ’14*. [online] doi:10.1145/2647868.2654940.

[15]

Vairavasundaram, S., Varadharajan, V., Vairavasundaram, I. and Ravi, L. (2015). Data mining-based tag recommendation system: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(3), pp.87–112. doi:10.1002/widm.1149.

[16]

Zhao, J. and Liu, Y. (2019). APIUaaS: a reference architecture for facilitating API usage from a data analytics perspective. *IET Software*, 13(5), pp.466–478. doi:10.1049/iet-sen.2018.5355.

[17]

Xiong, W., Lu, Z., Li, B., Hang, B. and Wu, Z. (2018). Automating smart recommendation from natural language API descriptions via representation learning. *Future Generation Computer Systems*, [online] 87, pp.382–391. doi:10.1016/j.future.2018.05.006.

[18]

Tsaku, N.Z. and Kosaraju, S. (2019). Boosting Recommendation Systems through an Offline Machine Learning Evaluation Approach. *Proceedings of the 2019 ACM Southeast Conference*. doi:10.1145/3299815.3314454.

[19]

Oramas, S., Nieto, O., Sordo, M. and Serra, X. (2017). A Deep Multimodal Approach for Cold-start Music Recommendation. *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. doi:10.1145/3125486.3125492.

[20]

Lee, W.-P., Chen, C.-T., Huang, J.-Y. and Liang, J.-Y. (2017). A smartphone-based activity-aware system for music streaming recommendation. *Knowledge-Based Systems*, 131, pp.70–82. doi:10.1016/j.knosys.2017.06.002.

[21]

Si, Y. (2021). Piano music recommendation algorithm based on deep learning neural network. *2021 4th International Conference on Information Systems and Computer Aided Education*. doi:10.1145/3482632.3482739.

[22]

Kim, H.-G., Kim, G.Y. and Kim, J.Y. (2019). Music Recommendation System Using Human Activity Recognition From Accelerometer Data. *IEEE Transactions on Consumer Electronics*, 65(3), pp.349–358. doi:10.1109/tce.2019.2924177.

[23]

Xu, D., Dai, Y. and Dai, W. (2019). Smart recommendation of personalized cloud music service based on mental model. *Proceedings of the ACM Turing Celebration Conference - China*. doi:10.1145/3321408.3322637.

[24]

Hu, W.-H., Tang, S.-H., Chen, Y.-C., Yu, C.-H. and Hsu, W.-C. (2018). Promotion Recommendation Method and System Based on Random Forest. *Proceedings of the 5th Multidisciplinary International Social Networks Conference on - MISNC '18*. doi:10.1145/3227696.3227708.

[25]

Zhao, S. (2021). Popular Song Recommendation Program Based on Machine Learning Algorithm. *2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture*. doi:10.1145/3495018.3495478.

[26]

Hong, D., Li, Y. and Dong, Q. (2020). Nonintrusive-Sensing and Reinforcement-Learning Based Adaptive Personalized Music Recommendation. *Proceedings of the 43rd International*

ACM SIGIR Conference on Research and Development in Information Retrieval. doi:10.1145/3397271.3401225.

[27]

Zangerle, E., Pichl, M. and Schedl, M. (2018). Culture-Aware Music Recommendation. *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. doi:10.1145/3209219.3209258.

[28]

Chaves, P.D.V., Pereira, B.L. and Santos, R.L.T. (2022). Efficient Online Learning to Rank for Sequential Music Recommendation. *Proceedings of the ACM Web Conference 2022*. doi:10.1145/3485447.3512116.

[29]
Wang, X., Wang, Y., Hsu, D. and Wang, Y. (2014). Exploration in Interactive Personalized Music Recommendation. *ACM Transactions on Multimedia Computing, Communications, and Applications*, [online] 11(1), pp.1–22. doi:10.1145/2623372.

[30]
Knees, P. and Schedl, M. (2015). Music Retrieval and Recommendation. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. doi:10.1145/2766462.2767880.

Chapter 1,3,4,5,6 and 7 references

[1]

Computer Science. (n.d.). *Definition and History of Recommender Systems*. [online] Available at: https://international.binus.ac.id/computer-science/2020/11/03/definition-and-history-of-recommender-systems/.

[2]

Boehmer, J., Jung, Y. and Wash, R. (n.d.). [online] Available at: https://rickwash.com/papers/recsys-encyclopedia.pdf.

[3]

Roy, A. (2020). *Introduction To Recommender Systems- 1: Content-Based Filtering And Collaborative Filtering*. [online] Medium. Available at: https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421.

[4]

SearchMobileComputing. (n.d.). *What is Google Firebase? - Definition from WhatIs.com*. [online] Available at: https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase.

[5]

theproductiveengineer.net. (n.d.). *Detailed Guide to Writing JavaScript in Atom - The Productive Engineer*. [online] Available at: https://theproductiveengineer.net/detailed-guide-to-writing-javascript-in-atom/ [Accessed 1 Sep. 2022].

[6]

www.javascript.com. (n.d.). *Free JavaScript training, resources and examples for the community*. [online] Available at: https://www.javascript.com.

[7]

Analytics Vidhya. (2020). *Distance Metrics | Different Distance Metrics In Machine Learning*. [online] Available at: https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/

[8]

Client-server Architecture Definition. (n.d.). [online] Available at: https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/ClientServer.pdf.

[9]

AltexSoft. (2019). *Web Application Architecture: How the Web Works*. [online] Available at: https://www.altexsoft.com/blog/engineering/web-application-architecture-how-the-web-works/.

[10]

Techopedia.com. (n.d.). *What is Active Content? - Definition from Techopedia*. [online] Available at: https://www.techopedia.com/definition/4847/active-content [Accessed 1 Sep. 2022].

[11]

Adam, J. (n.d.). *What is Agile software development? - K&C*. [online] Available at: https://kruschecompany.com/agile-software-development/.

[12]

explorable.com. (n.d.). *Steps of the Scientific Method - The Stages of Scientific Research*. [online] Available at: https://explorable.com/steps-of-the-scientific-method.

[13]

Sreenivasa, S. (2020). *Radial Basis Function (RBF) Kernel: The Go-To Kernel*. [online] Medium. Available at: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a.

[14]

www.grandviewresearch.com. (n.d.). *Recommendation Engine Market Report, 2021-2028*. [online] Available at: https://www.grandviewresearch.com/industry-analysis/recommendation-engine-market-report [Accessed 1 Sep. 2022].

[15]

Brownlee, J. (2020). *Introduction to Dimensionality Reduction for Machine Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/.

[16]

stattrek.com. (n.d.). *Covariance Matrix*. [online] Available at: https://stattrek.com/matrix-algebra/covariance-matrix.