Effective Use Of Mlops In Music Recommendation System

MSc In Cloud Computing

Pradeep Manoharan Susila
Student ID: x21121141

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Pradeep Manoharan Susila
Student ID:	X21121141
Programme:	Msc in cloud computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Vikas sahni
Submission Due Date:	15/12/2022
Project Title:	Effective use of MLOPS In music recommendation system
Word Count:	6012
Page Count:	17

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

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

Signature:	Pradeep Manoharan Susila
Date:	15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

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

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

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

Effective Use of ML ops in Music Recommendation System

Pradeep Manoharan Susila

X21121141

Abstract

Users' preferences such as ratings, only give unidimensional data, but the reasons for users' preferences might be tied to a variety of different attributes of an object. Here, item refers to things like categories or songs. We can determine the user's interest by examining comments generated by the user. Here, we're using the recommendations system to create a music recommendation model. Using the Spotify dataset, we extract collaborative and content-based elements to determine the user's listing pattern. To address the shortcomings of the standard recommendation system, we added ML ops to this system. By incorporating this methodology into the prior system, we can avoid manually training the data whenever new data enters the system, which is time-consuming. However, with the new system that is being proposed, when fresh data is introduced into the system, the mops will retrain the data without the need for human interaction. We will compare ML ops accuracy to the standard technique in this research. Then, when the user's tastes vary over time, we'll observe how the recommendation algorithm dynamically promotesmusic to them.

1 Introduction

In recent years, recommendation systems have been widely developed to alleviate the overload of data and to give personalised suggestions of items (for example, movies, books, music, and news). Nowadays, recommendation algorithms often seek to learn users' interests based on their item choice histories (sometimes utilising user evaluations on music) and make suggestions based on a network of similar individuals who share a common interest on the same items. This recommendation method has already been used in a variety of disciplines. For example, social networking services like Instagram employ a recommendation algorithm to propose buddies to users. Amazon Music and YouTube Music employ the very same recommendation engine to propose songs based on the user's search history or interests. Given this important basis, our goal is to develop a recommendation system for users in response to possible point of interest selections they may make.

1.1 Research Questions and Objectives

The study is being conducted with the goal of applying new methodologies that are performance driven and will yield dependable data.

The research question is: How effectively Mlops can increase the Accuracy of the music recommendation system and auto retrieval of the new data. there will be two types of recommendation systems that are often used: collaborative recommendation systems and content-based recommendation systems. Collaborative filtering systems collect music ratings or suggestions, identify patterns amongst users based on their ratings, and deliver new suggestions based on inter user comparisons. A content-based recommendation engine learns from a fresh user's account based on the qualities found in previously rated music. It is basically a keyword-based recommender system, with the keywords used to describe the music. This difficulty can be made easier by optimization techniques that can determine a user interest and identify other comparable users and/or music thatfit his/her selections.

We want to create a recommendation system that will allow us to generate cutting-edge music recommendations for assist users by using learning algorithms to create a hybrid model of customer evaluations and ratings. In this thesis, we chose to alter the traditional recommendation by including ML ops into the algorithm. In the old technique, they must train the data every time fresh data enters the system, which takes a long time and prevents them from making dynamic recommendations when the user's interests change. In this paper, we will examine how efficiently the ML ops can propose music to the user when the user's interests vary dynamically

2 Related Work

2.1 Audio Classification in recommendation system:

In this music genre categorization of audio signal, George et al. They classified audio signals in this work using timbral texture characteristics, spectral roll off, spectral flux, and rhythmic content features. They also attempted manual categorization of audio sampleswith the assistance of students, achieving 53per when hearing 250ms samples and 70perwhen hearing 3seconds samples. Tzanetakis and Cook (2002) However, they had failed to consider the range and complexity of the genre. They did not include emotion and vocalstyle analysis for improved categorization.

According to this study, music preferences had been isolated from the content-based recommendation system. The biggest issue with this sort of equipment is the cold start. They created a technique to detect these features or elements in music's acoustic content. The second issue is that the previous recommendation algorithm cannot handle songs with no ratings. They also did not categorise the music by genreSoleymani et al. (2015). The biggest disadvantage is that the underestimated songs did not have a lot of meta data. .

2.2 Recommendation system using machine learning:

They constructed a recommendation system using machine learning, but they ran into a number of issues, such as cold start issues with the content-based system.Fanca et al. (2020) They gathered information about the users' interests and interactions. Initially, they were running tests using the manual approach, whereas they had only evaluated only with collaborative filtering method, which they conducted multiple times to achieve better accuracy, and they also attempted to do in Azure, but they did not hold the users' related information and interests, and they were not connected this with the recommendation system.

Generated an effective machine learning-based recommendation for book selection, however they only used one approach, collaborative filtering, they took the results for accuracy, and they had to train the data every time fresh data enters the system.Kommineni et al. (2020) The precision had been combined with the efficiency of the normal distribution. They were not compared the accuracy of content and hybrid, and they did not set up the ml pipeline, which is a serious flaw.

A music recommendation system that used a dynamic k-means clustering technique. The fundamental issue was that people downloaded songs from multiple websites, however some websites just offer personalised suggestion services.Kim et al. (2007) They had solely used content-based and collaborative recommendations; initially, they were analysing the music's properties and then the dynamic k means algorithm before building the suggestion list. The accuracy was determined by combining the results of all users' listing patterns. They did not specify the data cleaning procedure, and when fresh data arrives, they must train the algorithm. This is the system's biggest disadvantage.

2.3 Content recommendation system:

Music recommendation based on content using underlying music preference structure. The biggest issue with this strategy was that when a new user joins the system, it was impossible to propose anything because there were no trace of the individual.Soleymani et al. (2015) The systems in the first group employ some kind of music similarity measure to solve a query-by-example issue, whereas the systems in the second category learn music preferences from the user profile. They employed acoustic characteristics and attribute learning in this case. The biggest disadvantage of the method was that they only have categorization for part of the songs and no classification for the others.

They developed a music recommendation system based on content and a collaborative approach, therefore eliminating the cold start problem. They used the customers' previous interests and behaviours to provide music recommendations using a content-based technique.Darshna (2018) subjective characteristics of music Speechiness, loudness, acoustiness, and so on Finally, they entered all of the data into the database. When the attribute meets the user's interest, the music will be recommended. The main disadvantage of this system is that they must train the new entering data, and when a new user arrives, the system struggles to recommend. This system is unable to carry out the sequential approach.

The issues encountered in this article were that whenever a new song was released, they failed to promote it to the user. To address this difficulty, they looked at the user's listening history and demographics. They also made tunes that consumers loved and ones that users despised.Chen et al. (2021) To overcome the cold start problem in their system, they built an audio branch and a user branch. The issue with this system is that it did not have a solution for offline users and had not satisfy the expectations for major update songs.

2.4 Collaborative recommendation system:

The collaborative filtering for music recommendation system, based to this principle. We investigated many metrics to evaluate the similarity of users and objects, such as Euclidean distance, cosine metric, Pearson correlation, and others.Shakirova (2017) Finally, we evaluated several assessment measures that describe the recommendation system's efficiency. They used a similarity measure, a scoring function, ranking aggregation, and assessment metrics to determine the efficiency of the system, however they only had used system and did not compare it to the others.

2.5 Hybrid Recommendation System:

They tried a novel way despite utilising the comparison of the two users since the end outcome will have some variances, so they may extract some crucial data from each of the user by using historical data resource to represent user engagement and to promote the similarity resource. The new algorithm was then compared to the classic collaborative filtering algorithm.Yang et al. (2016) They separated the training data into two halves, with 20 percent of the data buried in the test data and the remaining 80 percent utilised to create the recommendation result.

3 Methodology

3.1 Data set:

Spotify Dataset which includes genuine music ratings, user, and user review data from throughout the world. The key rationale for adopting the above dataset for our study is that it has a large amount of data and allows us to experiment with a variety of musical genres and user behaviour. The dataset was given in JSON format and then was converted to CSV in places based on our needs. For our investigation, we used a different users Spotify dataset. The major purpose for collecting this data set is because each user has a different musical taste and certain types of music they wish to hear on a regular basis, and they don't like some songs to recognise and offer to them.

3.2 Recommendation Models

3.2.1 Collaborative filtering:

Collaborative filtering algorithms are used to make predictions about a user's interests by matching preferences from several other users. The different types of collaborative filtering include memory-based collaborative filtering and model based collaborative filtering. For this paper, we have used model-based collaborative filtering. The model-based recommendation involves building a model based on the ratings. Here, we extract information from the rating dataset and use that as a" model" to provide recommendations without having to use the entire dataset every time. This methodology offers the benefits of both speed and scalability. SVD++ is an enhancement of the SVD method that takes into consideration implicit ratings. In this context, an implicit score is the fact that user u rated item I independent of the rating value. As a result, the SVD++ model includes SVD-based implicit feedback information; that is, it appends a factor vector for each item, and those item factors are utilised to express an item's characteristics regardless of whether it has been assessed or not. The user's factor matrix is then modelled in order to generate a more reliable user bias.

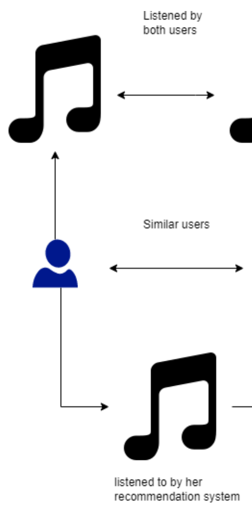

Figure 1: Collaborative filtering

3.2.2 Content based model

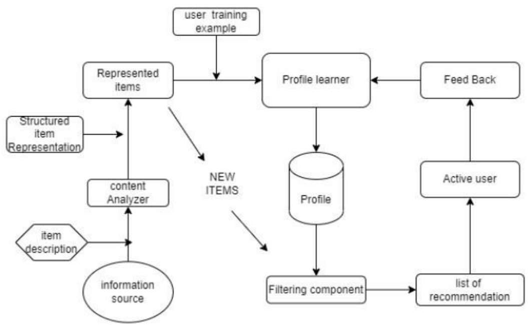

Figure 2: Content Based Model

Content-based models are commonly seen as an extension and expansion of data filtering research. The objects in this model are determined by their associated search. A content-based approach recognises the user's interests based on the user's current and previous rated items. Items can refer to music in this context. This will function similarly to a keyword-specific recommender system, with keywords specifying the music. This recommendation system employs an algorithm to propose comparable things that the user has previously enjoyed or is presently considering.

3.2.3 Hybrid based model

Among the most common recommendation systems are content-based filtering and collaborative filtering. Both of these recommendation methods have advantages and downsides. These algorithms' key issues are cold start and data sparsity. To overcome each of these shortcomings, we may combine the two models to produce a superior recommendation

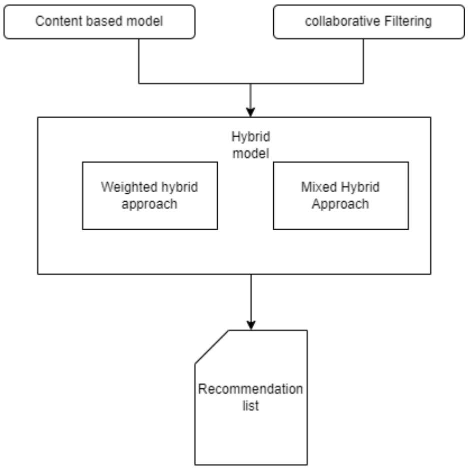

Figure 3: Hybrid Based Model

system. The hybrid recommendation refers to the merging of two or more recommendation systems. By integrating the two techniques, we may avoid the cold start problem by offering recommendations from a content-based model and provide improved recommendations for old users using collaborative filtering.

3.3 MLOPS Proposed system

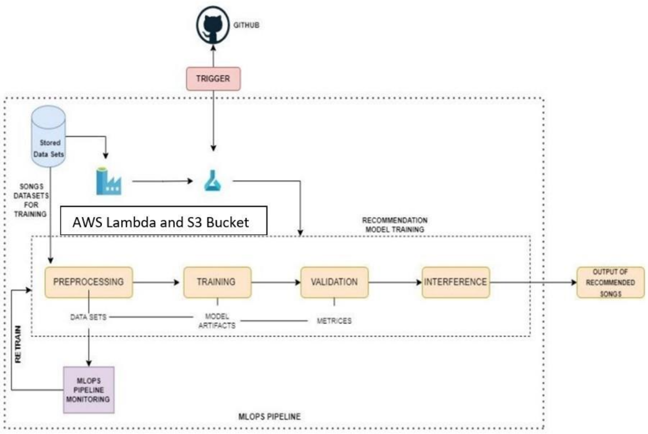

Figure 4: Pipeline Model

MLOPS is an artificial intelligence-infused machine learning operation. AI systems will benefit from MLOPS even during creation and deployment processes. MLOPS is taken from the DevOps ideas used in the development area and simplifies the development lifecycle in terms of tracking and maintaining the process cycle. This technology will be used in this research to automate the recommender systems model's procedure. This will mostly concentrate on machine learning advancement The term "MLOps" integrates "machine learning" as well as the continuous development process of DevOps to ensure that the deployment of a machine learning system is as trustworthy and effective as feasible. MLOps include creating automated workflows, controlling the pipeline, and reporting on pre-defined business metrics. This may be used to create a machine learning pipeline including complete automation and model creation. This allows us to keep an eye on the entire model.

This MLOPS primarily consists of two critical stages: Continuous Integration (CI) and Continuous Delivery (CD). This basically automates the process of establishing a model of development. From the analysis, we will divide the phases into continuous integration for testing the metrics for the trained model, because when the trained metrics have excellent accuracy and a flawlessly anticipated model, it will proceed to the 2nd phase, Continuous Delivery. This allows us to keep an eye on the entire model.

This MLOPS primarily consists of two critical stages: Continuous Integration (CI) and Continuous Delivery (CD). This basically automates the process of establishing a model of development. From the analysis, we will divide the phases into continuous integration for testing the metrics for the trained model, because when the trained metrics have excellent accuracy and a flawlessly anticipated model, it will proceed to the 2nd phase, Continuous Delivery. In this step of continuous delivery, we will identify the deployment of the trained model. Services may leverage the deployment to get our model online. This is a pipeline of automated processes that will be used in my project.

4 Design Specification

4.1 Data Transformation Layer

As the name of this stage implies, this part of the process includes the interpretation of the data, which is an essential component of the total research study. During this stage, the layer will be responsible for managing all processes connected to data. It consists of three fundamental processes, which are the collection of data for the study, the analysis of the raw data that was acquired, and the discovery of structures to understand more about it.

4.2 Modeling Layer

As its name suggests, this phase involves the interpretation of the data, which is crucial to the research process as a whole. This layer's current job is to oversee everything having to do with data processing. The study's three main phases are data gathering, data analysis, and structure finding.

4.3 Data Storage and Archival

Having access to reliable data is essential for the growth of any business. It's needed for all sorts of things, including research, planning, and assessing market trends. Therefore, it is crucial to keep the data updated and stored safely. The data used in the modeling process is kept on a compatible database platform, such as the cloud, where it can be accessed whenever it is needed by the industry or business. Data used in the process is archived and stored on a remote cloud platform, which takes care of the archiving and storage needs. The data for this study was stored in the cloud using the AWS sage maker, s3 bucket, etc..

4.4 MLOPS PIPELINE:

In this study, the MLOPS Pipeline architecture will be utilised to totally construct an automated process flow for Music Recommendation System. This automated process flow will aid in continually monitoring the pipeline for changes in code, loss of health, prediction accuracy, and so on. We will divide the music system recommendation pipeline into three phases in this study. The steps are as follows: Source, Continuous Integration, and Continuous Delivery. Each stage will be important in the development of this project. In the following paragraphs, we will go through all three crucial steps. First, we'll go through the Source Stage, where we utilise a repository to store all of the code and data acquired for our music recommendation engine. We will utilise the GitHub repository for the source stages, which is an open-source platform where we may keep the source code for our application. And for data, we utilise a database to hold the datasets that will be used to train the model. This source stage's data and code are utilised to develop our entire music system model. When data is acquired, it is stored directly in just this Source stage for use in the process. The next step is critical since it is where the key procedure will take place. We will do Continuous Integration at this point. This Continuous Integration stage will include several critical crucial validations, with the Trigger feature playing a significant part. We complete the music suggestion instruction at this point. Pre-processing, training, validation, and interference of a flawless music recommendation model are all part of the training model. For example, if we make any code changes, this stimulates the entire model training process and assesses and checks the accuracy. We may define the accuracy limit so that the model building process continues until that accuracy is reached. As a result, this aids in the maintenance of an automated process.

Furthermore, we may provide alternative model training by dividing the phases, compare all the accuracy, and select the best model based on the achieved accuracy. When it comes to music prediction, testing such as parameter tweaking may be done to increase the forecast's accuracy for the user. Furthermore, during the CI stage, we may do a quality check on the code and assign test cases to be true or false. This Continuous Stage will also provide essential features such as Optimize AUC and K-mean prediction. Finally, only this trained suggested output will be transmitted to the next stage when all criteria have been evaluated and certified to green.

the music recommendation system that is already in place has been trained through a variety of manual processes, including data cleansing, pre-processing, and the creation of the pipeline. They have to manually train the data, which is a complicated process because they have to develop it to execute the specified task. This process is required whenever fresh data is brought into play. When they have more knowledge on the user, they will be able to achieve a higher level of accuracy. The accuracy will be poor if they did not obtain the necessary information about the user. in addition, there could be some issues with the two different filtering methods, namely with the content-based strategy. Both this strategy and the way of using a content-based system have an issue with their "cold starts." The primary issue is that the occurrence of the problem is contingent upon the introduction of a new user into the system.

5 Implementation

5.1 Programming language

This analysis makes use of Python, a programming language. This programming language is both high-level and straightforward. Python's syntax is straightforward and easy to grasp. Python simplifies the process of integrating new systems and programs. From modeling to visualisation, it covers it everything with its extensive library.

5.2 Data Cleaning

5.2.1 Null Value Treatment

Both IsNull() and sum() Python methods were used to check for null values in the data-set, and the results showed that there were neither null nor Na values included in the information. The dataset has had any instances of null values removed.

5.2.2 Duplicate Treatment

In addition, the identical entries in the dataset may be located with the help of the function duplicated (). There were many occurrences of the same value, which accounted for around 2–4Percent of the total variation. There are very few duplicate records in the dataset; as a result, removing those values will not have any impact on the integrity of the data or the results. As a result, the entries that were found to be redundant were eliminated from the dataset.

5.3 Existing model

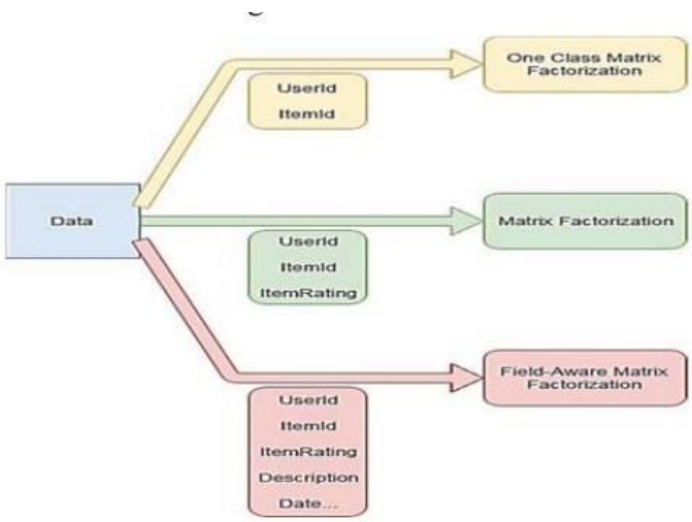

Figure 5: Existing model

the music recommendation system that is already in place has been trained through a variety of manual processes, including data cleansing, pre-processing, and the creation of the pipeline. They have to manually train the data, which is a complicated process because they have to develop it to execute the specified task. This process is required whenever fresh data is brought into play. When they have more knowledge on the user, they will be able to achieve a higher level of accuracy. The accuracy will be poor if they did not obtain the necessary information about the user. in addition, there could be some issues with the two different filtering methods, namely with the content-based strategy. Both this strategy and the way of using a content-based system have an issue with their ”cold starts.” The primary issue is that the occurrence of the problem is contingent upon the introduction of a new user into the system.

5.4 Proposed Model

The conventional model of music suggestion has been replaced with a new one that is very different from the old one. For the purpose of automating the procedure, we have integrated the music recommendation system with the cloud. We are use the AWS sage maker, s3 bucket, and lambda in Amazon Web Services (AWS). By putting in place the sage maker, we are uploading all of the CSV files and the ipynb file in the notebook. After that, all of the data will be saved in the s3 bucket. After that, by putting in place the lambda function, whenever new data enters the system, the lambda will automatically trigger, and we will be able to get all of the logs. Through the use of the cloud watch, it will be possible to record what kind of data has been uploaded, as well as when it has been uploaded, and access all of the associated information. If the newly supplied data from Spotify signifies that it will automatically train the data, then it will deploy in the system. We are using the information that was provided by Spotify, which comprises the listening habits of various people. We are going to take that data, and then we are going to clean the data. During that cleaning procedure, any data that is deemed undesirable will be deleted. Afterwards, we will get rid of the data that has a null value. We are able to circumvent the issue with the system’s cold start by putting this into action.

6 Evaluation

6.1 Sage Maker

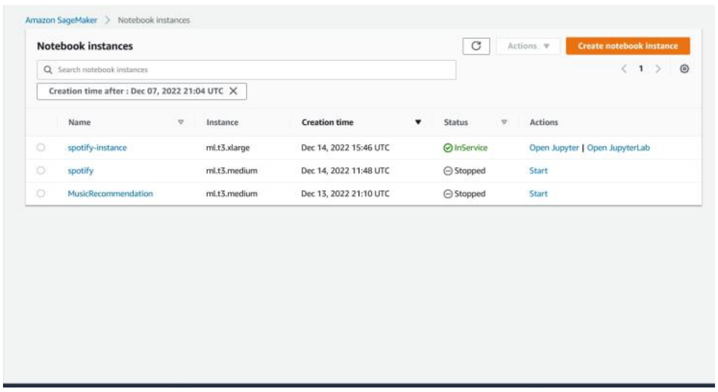

Figure 6: Sage Maker

A completely managed service for machine learning, Amazon SageMaker may be found on the Amazon website. We are able to swiftly and efficiently develop machine learning models with SageMaker, as well as train them, and then instantly deploy them into a production environment. It eliminates the need for us to operate servers by providing an integrated Jupyter writing notebook session, which enables quick and simple access to your data sources for the purposes of exploration and analysis. In addition to this, it offers standard machine learning methods that have been fine-tuned for effective operation even when confronted with extremely big datasets in a distributed setting. SageMaker offers versatile distributed training alternatives that may be adapted to your particular workflows.

6.2 S3 Bucket

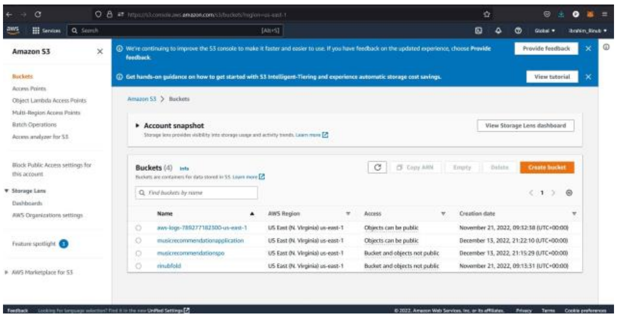

Figure 7: S3 Bucket with spotify dataset

We have been storing everything in Amazon's S3 bucket, which is a storage solution that is known for its scalability, data availability, security, and performance. Amazon S3 allows you to store an unlimited quantity of data and access it at any time and from any location you want. The Spotify data has been saved in an S3 bucket that we control. We are able to determine the type of data that has been added to the system, and the model will be automatically trained in accordance with it. Additionally, the lambda function will be automatically called whenever any data is added to the system.

6.3 Lambda Function

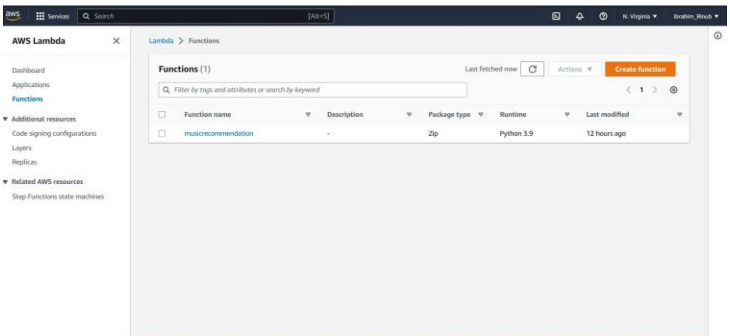

Figure 8: Function Dashboard

A serverless computing service, AWS Lambda executes your code in response to events and takes care of the underlying resource management for you automatically. When the

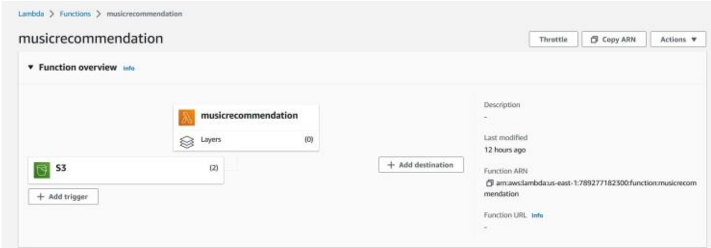

Figure 9: Function overview

new data file is loaded into the system, this will cause an update to be triggered. Code is automatically executed by AWS Lambda in response to a number of events, including HTTP requests sent through Amazon API Gateway, changes made to objects stored in Amazon Simple Storage Service (Amazon S3) buckets, table updates sent through Amazon DynamoDB, and state transitions sent through AWS Step Functions. Lambda manages all of your computing resources and runs our code on high-availability infrastructure. It also handles all of the management of our own infrastructure. This covers the management of servers and operating systems, the provisioning of capacity and the automatic scaling of that capacity, the distribution of code and security patches, and the monitoring and logging of code.

6.4 Cloud Watch

using cloud watch we can monitor and manage the system data and insights about the data. using this we can get the performance and operational data all in an single place. we can also place an alarm when an undefined data enters the system. all the process logs can be viewed here like whether the new data has been trained or not and the new data is deployed into the ml model or not all the insights can be gathered here.

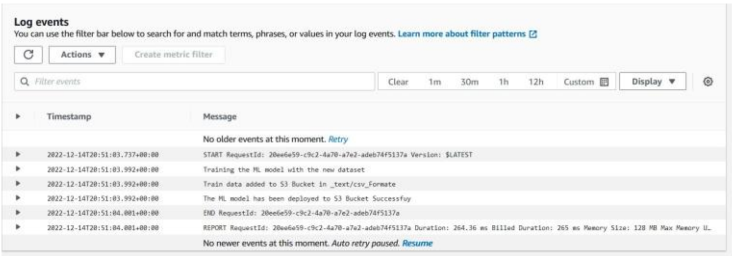

Figure 10: Cloud Watch Logs

6.5 Metrics

Top-N recommender systems show consumers a prioritised list of N things in which they are likely to be interested in order to encourage purchases and views. We want to utilise suggestions to mimic the real world rather than merely anticipate user ratings. After examining prior metrics used to evaluate recommender systems, we decided to analyse our model using Mean Reciprocal Rank (MRR) and Hit rate.

6.5.1 Mean Reciprocal Rank

The user is unconcerned with the rating of the music that has been recommended to him rather, the user is concerned that the recommended music is relevant to his/her preferences. Given this, we must assess the system based on its rating of the suggested list instead of its accuracy. As a result, we can employ (MRR) for assess the recommendation model's success.

If Q is the number of individuals and ranking is the position of the relevant item in a user's suggested list. For each user, a user-relevant matrix with ratings for all businesses previously examined is created in order to evaluate the model. The features user id, songs, rating, and the applicable tag will be included in the user-relevant matrix. Based on various circumstance, the corresponding label will just be value of 0 or 1.

- 0 if indeed the formed as a result to the song which means it is lower than the user's average rating.
- 1 if the songs rating is greater or comparable to the user's average rating.The song ids which has an value of 0 are eliminated from the dataset, so that the liked song only present in the dataset. The suggested list of an user is compared to the hold out set to identify that any interest recommended is present in the hold out set.
- If there is a match, the inverted rank of the first incidence of the song for that user is determined.
- If there isn't any match, the user's reciprocal rank is 0. Steps 1 and 2 are performed for N users, and the mean value of the reciprocal rank for all N users is determined, yielding the MRR score for that model.

6.6 Song Classification by Hit Music

In our music recommendation system, we have divided the popular songs that users prefer into two categories: hit songs and non-hit songs. With the help of this categorization, we are able to increase the accuracy of our users' lists of hit songs, which are now complete.

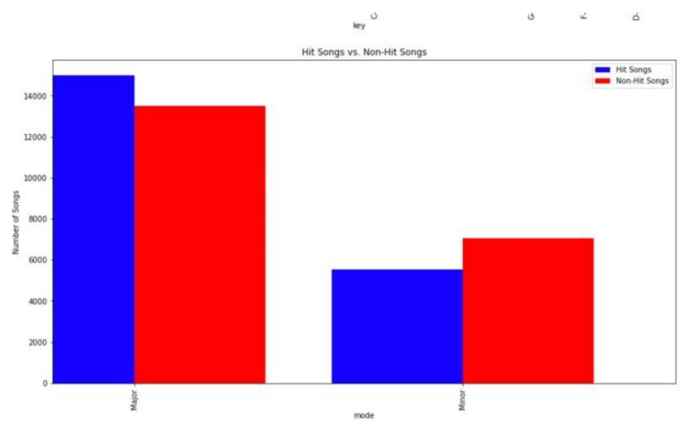

Figure 11: Hit song By the user

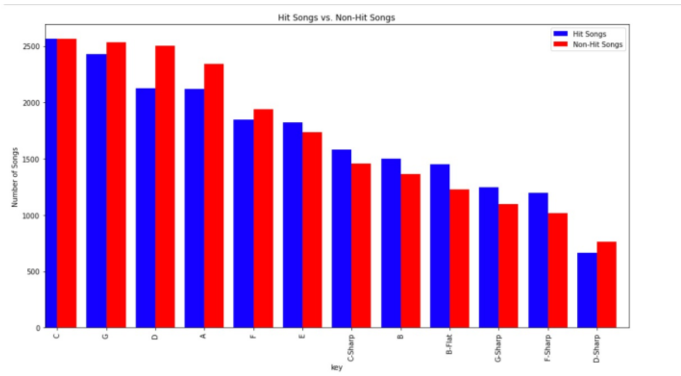

Figure 12: Total hit songs

6.7 Song classification by Specification

the danceability of the song, the energy of the song, the loudness of the song, the speechiness of the song, the instrument, the liveness of the instrument, the valence of the instrument, the tempo of the instrument, the duration of the instrument in milliseconds, the sections, by this we can separate all of the songs, and the listening patterns of the users, we can suggest according to them.

```
danceability :  T_stat:  74.7901595835779 / P_value:  0.0

energy :  T_stat:  36.49114636517893 / P_value:  5.987111564095718e-287

loudness :  T_stat:  60.51944378914344 / P_value:  0.0

speechiness :  T_stat:  -8.285943324340233 / P_value:  1.206807801225774e-16

acousticness :  T_stat:  -51.46355964028172 / P_value:  0.0

instrumentalness :  T_stat:  -90.50610383659532 / P_value:  0.0

liveness :  T_stat:  -10.443862406965835 / P_value:  1.6824912751669193e-25

valence :  T_stat:  52.603756520146916 / P_value:  0.0

tempo :  T_stat:  6.622784650102702 / P_value:  3.5682852808286016e-11

duration_ms :  T_stat:  -15.007206361516776 / P_value:  8.980976712179482e-51

chorus_hit :  T_stat:  -9.419065151555587 / P_value:  4.779192361662327e-21

sections :  T_stat:  -12.185826684530705 / P_value:  4.235949580734437e-34
```

Figure 13: Song Specification

6.8 Song Positive and negative rate

if the user liked the song that we suggested, then the liked song will be saved to the s3 bucket, and if the user did not like the song that we suggested, then the songs will be removed from the database. the positive and negative rate have been classified in this song so that the song can be classified according to it. the song has been shown to the user. When a new song is made available to the public, the relevant data will be uploaded to the system in an automatic fashion. Simultaneously, the model will undergo automated training, and once it is complete, it will be deployed into the system. Once more, the procedure will carry on without any intervention for the user.

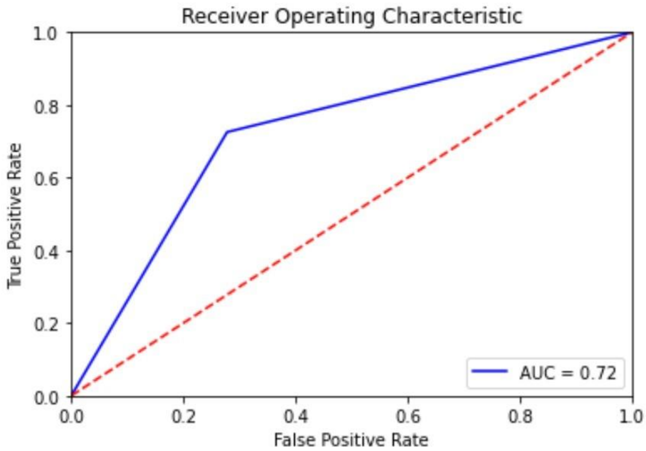

Figure 14: False and Positive Rate

6.9 Random Forest Confusion Matrix

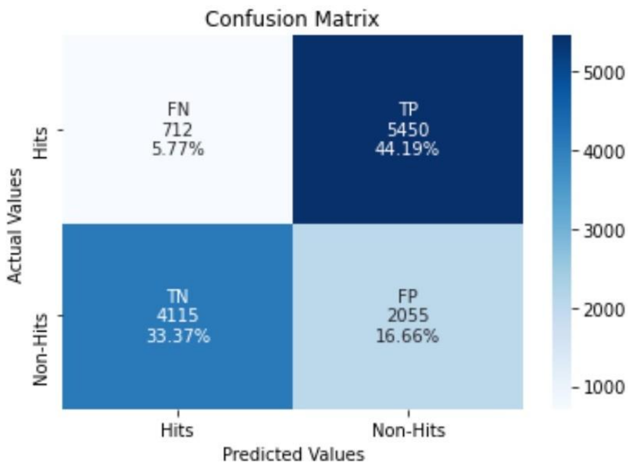

Figure 15: Random Forest Confusion Matrix

using The confusion matrix stores the results of many tests to see how well the model can classify data. A class vote for a row in the out-of-bag data is the anticipated class for the row based on the information from a single tree in the forest for a specific tree in the forest. Using this strategy, we have taken the values that were predicted and the values that really occurred, as well as the data that we have regarding the hits and the non-hits that we have done. When we classified the data using Random Forest, we were able to get an accuracy score of 86.1

	precision	recall	f1-score	support
Hits	0.85	0.67	0.75	6170
Non-Hits	0.73	0.88	0.80	6162
accuracy			0.78	12332
macro avg	0.79	0.78	0.77	12332
weighted avg	0.79	0.78	0.77	12332

Figure 16: Evaluation Metrics Score Random forest

6.10 Discussion

In order to find a solution to the study issue of how Mlops will incorporate the Retrain model into the system and increase the accuracy of the music recommendation system, the research makes use of the dataset that is provided by Spotify. Several methodologies were utilized in the selection process for the research question. the data are trained by the machine learning models and utilised. We have provided the answer to the issue about the chilly start. The correctness of the new suggested model in comparison to the older model has been provided in the table that was just described. The effort involved in processing the data and handling the null value data are two of the most significant problems that were encountered while carrying out the research. It was an essential step in the process of creating the study and bringing it to fruition. When the results of the research question as a whole are considered, it is clear that all of the models have made significant advancements in terms of their performance and their ability to deal with the research problem. Taking into consideration how KNN was implemented and how well it performed on the same dataset, it is clear that KNN may be considered as an additional choice for the identification of fraud in any given circumstance. According to the findings, ANN is the most important neural network technology that should be evaluated for use in the future for a variety of detecting jobs.

7 Conclusion and Future Work

The purpose of the research was to develop a better music recommendation model that can automatically train itself with fresh data whenever it is added to the system. In order to address the issue of an unevenly distributed dataset, the KNN was applied. The findings of the models point to a good performance on the part of the models in addressing the problem with the study, and the implementation of this has led to an improvement in the level of accuracy. The initial phase of applying models with their default settings is followed by the implementation of a KNN model to improve model performance using the mean reciprocal rank approach. This step contributes to an improvement in the overall performance of the machine learning models. It is possible to draw the conclusion, as a closing point on the research, that the research was successful in achieving its goal by using a select number of the models that performed very well and assisted the study in achieving its objectives.

As a result of this future work, we will be able to scale this music suggestion by including the cluster into the sage maker. Additionally, we will be able to increase both the accuracy and the speed of the system.

References

Chen, K., Liang, B., Ma, X. and Gu, M. (2021). Learning audio embeddings with user listening data for content-based music recommendation, *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3015–3019.

Darshna, P. (2018). Music recommendation based on content and collaborative approach reducing cold start problem, *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1033–1037.

Fanca, A., Puscasiu, A., Gota, D.-I. and Valean, H. (2020). Recommendation systems with machine learning, *2020 21th International Carpathian Control Conference (ICCC)*, pp. 1–6.

Kim, D., Kim, K.-s., Park, K.-H., Lee, J.-H. and Lee, K. M. (2007). A music recommendation system with a dynamic k-means clustering algorithm, *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 399–403.

Kommineni, M., Alekhya, P., Vyshnavi, T. M., Aparna, V., Swetha, K. and Mounika, V. (2020). Machine learning based efficient recommendation system for book selection using user based collaborative filtering algorithm, *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pp. 66–71.

Shakirova, E. (2017). Collaborative filtering for music recommender system, *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 548–550.

Soleymani, M., Aljanaki, A., Wiering, F. and Veltkamp, R. C. (2015). Content-based music recommendation using underlying music preference structure, *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing* **10**(5): 293–302.

Yang, J., Yang, C. and Hu, X. (2016). A study of hybrid recommendation algorithm based on user, *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 02, pp. 261–264.