#### Major Project Report

#### On

# “Music Recommendation System”

***Submitted in partial fulfillment of the requirements for the award of the degree of***

## Master of Computer Applications

***By***

## Akshay Sharma

###### Enroll no. A50500718003

***Under the guidance of Under the guidance of***

**Dr. Charu Jain**

**Assistant Professor**

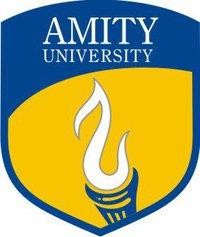
CSE Department

## 

**Ms. Aarti Chugh**

**Assistant Professor**

CSE Department



#### Amity Institute of Information Technology

**Amity University Haryana**

**May 2021**

**Amity Institute of Information Technology**

**Amity University Haryana**

**DECLARATION**

I, **Akshay Sharma** student of MCA (MASTER OF COMPUTER APPLICATIONS) hereby declare that the report entitled **“Music Recommendation System”** which is submitted to, Amity Institute of Information Technology, Amity University Haryana, in partial fulfillment of the requirement for the award of the degree of Masters of Computer Applications, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

**Date:01-05-2021**

****

**Akshay Sharma**

**A50500718003**

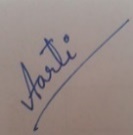
**2018-2021**

**Amity Institute of Information Technology**

**Amity University Haryana**

## Certificate

This is to certify that Akshay Sharma (Enrolment N0. A50500718003), student of MCA VIth semester, AIIT, Amity University Haryana, has done his Integrated Major Project entitled “Music Recommendation System” under my guidance and supervision. The work was satisfactory. He has shown complete dedication and devotion to the given work.



***Signature Signature***

**Dr. Charu Jain**

**Assistant Professor**

CSE Department

**Ms. Aarti Chugh**

**Assistant Professor**

CSE Department

***Date:* 01-05-2020**

###### Head

Department of Computer Science & Engineering Amity School of Engineering and Technology Amity University Haryana, Gurgaon



###### 

###### ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Dr. Charu Jain, Assistant Professor Department of CSE and Ms. Aarti Chugh Assistant Professor Department of CSE, Amity Institute of Information Technology, Gurugram, for their consistent guidance and unwavering support, collegiality and mentorship throughout this project. Without their thoughtful reassurance and careful supervision, this thesis would never have taken shape. Their encouragement and belief in me kept me enthusiastic and helped me perceive the work with new perception striving for excellence

I am thankful to my family whose value to me only grows with each passing day. They have helped me fight the highs and the lows and given me the strength to preserver especially through the times when the project seemed to be a mammoth task.

My earnest gratitude to the administration and staff of Amity Institute of Information Technology and all the other contributors to the study who have directly or indirectly supported and helped me.

Last but not the least I thank God Almighty.

**Akshay Sharma**

**(Enrolment no.: - A50500718003)**

# ABSTRACT

Nowadays lots of music industries like amazon music, wink music, gaana.com and Spotify are using recommender systems and the old-fashioned way of selling music has changed to a totally different cloud based. But the issue is there are lot of songs present in the cloud system. So, we need to classify all the songs based on different genres, artists locations, languages and the main goal is to classify these set of songs in accordance to the taste. Because user expects valuable return after the investment of time as well as money thereby we can attract a lot of customers by providing various valuable services of their interests for this project we are using cosine similarity algorithm. In this project we are going to use Collaborative filtering and Content-based models. In first part, Collaborative filtering we have categorized the songs on the bases of popularity by analyzing number of times user listens to songs. In second part we use Collaborative filtering. In collaborative filtering we use genres of songs which shows which type of song is this (rock, hip-hop etc.) on the bases of we recommend songs.

***Keywords****: Recommendation System, Genres, Cosine similarity, Collaborative filtering, Content-based filtering*

# List of Figures

# S.no. Name Page no.

1.1 Types of Recommendation engine------------------------------------------------------------2

2.1 Companies using recommendation------------------------------------------------------------6

4.1 Basic steps for Music Recommendation-----------------------------------------------------13

4.2 Graph show popularity of songs in data after losing data ---------------------------------14

4.4.1.1 Cosine similarity formula----------------------------------------------------------------------15

4.4.1.2 Cosine similarity graph-------------------------------------------------------------------------15

4.4.2.1 Euclidean Distance Formula--------------------------------------------------------------------16

4.4.3.1 Manhattan Distance Formula-------------------------------------------------------------------17

## 4.5.1 Result On the bases of Popularity (Cosine Similarity) ------------------------------------17

## 4.5.2 Result On the bases of Popularity (Euclidean distance) -----------------------------------17

## 4.5.3 Result On the bases of Popularity (Manhattan distance) ----------------------------------18

4.5.2 Results On the bases of Genres (content-based filtering)----------------------------------18

# List of Tables

# S.no. Name Page no.

# 2.1 Popular application domains of Recommendation system with filtering technique ----7

TABLE OF CONTENTS

# Name Page no.

[Declaration](#_bookmark0) i

[Certificate](#_bookmark1) ii

[Acknowledgment](#_bookmark2) iii

[Abstract](#_bookmark3) iv

[List of Figures](#_bookmark4) v

List of Tables vi

[Chapter 1 INTRODUCTION](#_bookmark5)

* 1. Different Recommendation Engines 1
  2. Machine Learning 3

1.2.1 Prediction in Machine Learning 3

[Chapter 2](#_bookmark15) BACKGROUND STUDY

* 1. Use of Recommendation System 5
  2. Objective 8

[Chapter 3](#_bookmark19) TOOLS AND TECHNIQUES

3.1 System Architecture 9

3.2 Operating System 9

3.3 Software and Library used 9

3.3.1 Python Programing Language 9

3.3.2 NumPy and Pandas. 10

3.3.3 Matplotlib 10

3.3.4 SciKit Learn 11

3.3.5 Seaborn 11

3.3.6Jupyter Notebook 12

[Chapter](#_bookmark19) 4 METHODOLOGY

* 1. Dataset 13

4.2Data Cleaning 14

4.3 Feature Extraction 14

4.4 Algorithm Used 15

4.4.1 Cosine Similarity 15

4.4.2 Euclidean Distance 16

4.4.3 Manhattan Distance 16

4.5 Result After Analyzing 17

[Chapter](#_bookmark19) 5 CONCLUSION AND FUTURE SCOPE

5.1 Conclusion 19

5.2 Future Scope 19

**References 20**

**Chapter 1**

**INTRODUCTION**

Music is an integral part of our life. We listen to music everyday as per our taste and mood. With the advancement and increase in volume of digital content, the choice for people to listen to diverse type of music has also increased significantly. Thus, the necessity of delivering the most suited music to the listeners has been an interesting field of research in computer science. One of the important measures to deliver the best music to listeners could be his/her personality trait. In this project, we aim to discover the impact of personality traits on the collaborative filtering (user to user) which is one of the most popular recommendation engines used today. In order to determine the personality of a person, social media like Facebook can be a useful platform where people express their views on different matters, share their opinions and thoughts. Such expressions of thoughts and opinions can be leveraged to study the personality traits of the person and hence use this information to try to enhance existing user to user collaborative filtering techniques for music recommendation.

**1.1 Different Recommendation Engine**

There are majorly six types of recommender systems which work primarily in the Media and Entertainment industry: **Collaborative Recommender system, Content-based recommender system, Demographic based recommender system, Utility based recommender system, Knowledge based recommender system**and**Hybrid recommender system.**

* **Collaborative Recommender System:** It’s the most sought after, most widely implemented and most mature technologies that is available in the market. Collaborative recommender systems aggregate ratings or recommendations of objects, recognize commonalities between the users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. The greatest strength of collaborative techniques is that they are completely independent of any machine-readable representation of the objects being recommended and work well for complex objects where variations in taste are responsible for much of the variation in preferences. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future and that they will like similar kind of objects as they liked in the past.
* **Content based Recommender System:** It’s mainly classified as an outgrowth and continuation of information filtering research. In this system, the objects are mainly defined by their associated features. A content-based recommender learns a profile of the new user’s interests based on the features present, in objects the user has rated. It’s basically a keyword specific recommender system here keywords are used to describe the items. Thus, in a content-based recommender system the algorithms used are such that it recommends users similar items that the user has liked in the past or is examining currently.
* **Knowledge based Recommender System:** This type of recommender system attempts to suggest objects based on inferences about a user’s needs and preferences. Knowledge based recommendation works on functional knowledge: they have knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a need and a possible recommendation.[9]
* **Hybrid Recommender System:** Combining any of the two systems in a manner that suits a particular industry is known as Hybrid Recommender system. This is the most sought-after Recommender system that many companies look after, as it combines the strengths of more than two Recommender system and also eliminates any weakness which exist when only one recommender system is used.



Fig 1.1: - Types of Recommendation Engines [12]

**1.2 Machine Learning**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, [an approach based on semantic analysis mimics the human ability to understand the meaning of a text](https://expertsystem.com/learning-center/technology/).[2]

Machine learning algorithms are often categorized as supervised, unsupervised or Reinforcement learning.

* **Supervised machine learning**: - It is applied when we have dataset and we try to learn from it and predict according to that. examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* **Unsupervised machine learning**: - It is used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
* **Reinforcement machine learning: -** It is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.[2]
* **Deep learning**: - Deep learning is a specific method of machine learning that incorporates neural networks in successive layers to learn from data in an iterative manner. Deep learning is especially useful when you’re trying to learn patterns from unstructured data. Deep learning complex neural networks are designed to emulate how the human brain works, so computers can be trained to deal with poorly defined abstractions and problems. The average five-year-old child can easily recognize the difference between his teacher’s face and the face of the crossing guard. In contrast, the computer must do a lot of work to figure out who is who. Neural networks and deep learning are often used in image recognition, speech, and computer vision applications.
  + 1. **Prediction in Machine Learning**

Machine learning algorithms in recommender systems are typically classified into two categories — content based and collaborative filtering methods although modern recommenders combine both approaches. Content based methods are based on similarity of item attributes and collaborative methods calculate similarity from interactions. Below we discuss mostly collaborative methods enabling users to discover new content dissimilar to items viewed in the past.

**Chapter 2**

**Background Study**

Nowadays social networking sites (such as Facebook, Twitter, etc.) have emerged as a substantial platform for applying RSs. These popular sites are considered to be the major source of information about people and hence becoming a great option to leverage novel and innovative approaches for the recommendation, leaving behind the old methods, to increase the accuracy.[5] The contextual information such as time, place, the emotion of people and groups in these social networking sites opens up a new avenue of recommendation known as contextual recommendation system. It also provides a good prospect to bring a dynamic essence in the recommendation. Seasonal marketing and conference recommendation are also emerging as considerable application areas in the context-aware recommendation.

**2.1 Use of Recommendation System**

Look back at your week Machine Learning algorithm determined what songs you might like to listen to, what food to order online, what posts you see on your favorite social networks, as well as the next person you may want to connect with, what series or movies you would like to watch.

Harvard Business Review made a strong statement by calling Recommenders the single most important algorithmic distinction between “born digital” enterprises and legacy companies. HBR also described the virtuous business cycle these can generate: the more people use a company’s Recommender System, the more valuable they become and the more valuable they become, the more people use them.

Looking at Netflix, for example, shows how crucial this is, as 80% of what people watch comes from some sort of recommendation. If we look at Amazon, 35% of what customers purchase at Amazon comes from product recommendations and at Airbnb, Search Ranking and Similar Listings drive 99% of all booking conversions.

* similar home listings (Airbnb, Zillow)
* relevant media, e.g., photos, videos and stories (Instagram)
* relevant series and movies (Netflix, Amazon Prime Video)
* relevant songs and podcasts (Spotify)
* relevant videos (YouTube)
* similar users, posts (LinkedIn, Twitter, Instagram)
* relevant dishes and restaurants (Uber Eats)

The “classic”, and still widely used approach to recommender systems based on **collaborative filtering** (used by Amazon, Netflix, LinkedIn, Spotify and YouTube) uses either User-User or Item-Item relationships to find similar content.

Amazon was probably the first company to leverage item-to-item collaborative filtering. When they first released the inner workings of their method in a paper in 2003, the system had already been in use for six years. Then, in 2006 Netflix followed suit with its famous Netflix Price Challenge which offered $1 million to whoever improved the accuracy of their existing system called Cinematic by 10%. Collaborative filtering was also a part of the early Recommender Systems at Spotify and YouTube. LinkedIn even developed a horizontal collaborative filtering infrastructure, known as Browse maps. This platform enables rapid development, deployment, and computation of collaborative filtering recommendations for almost any use case on LinkedIn.[6]



Fig 2.1: - Companies using recommendation system [13]

There are many domains that uses different recommendation system and having different filtering approach as shown in the given table.

Table 2.1: - Popular application domains of Recommendation system with filtering technique [4]

|  |  |  |  |
| --- | --- | --- | --- |
| S.no. | Application Domain | Filtering Approach used | Related Research papers |
| 1 | E-government | |  | | --- | | Collaborative | | Collaborative, hybrid | | |  | | --- | | Guo and Lu (2007) | | Wu et al. (2015) and Lu et al. (2010) | |
| 2 | E-library and E-learning | |  | | --- | | Content, Collaborative, Hybrid | | |  | | --- | | Cobos et al. (2013), Santos et al. (2014) | |
| 3 | E-tourism | |  | | --- | | Knowledge-based, collaborative | | Context-aware, Collaborative | | |  | | --- | | Ruotsalo et al (2013), Console et al. (2013) | | Xie et al. (2013) | |
| 4 | E-Resources | |  | | --- | | Content-based | | Collaborative | | |  | | --- | | Jinni (2017), Rotten tomatoes (2017), IMDB (2017), ACRnews (2017) | | FoxTnt (2017), Tastekid (2017), nano Crowd (2017),  Movielens (2017) | |

## 2.2 Aims and Objectives

This project aims to find the various songs so that if a user’s taste is similar to the other one we can recommend the songs of one to another on the basis of similar taste.

## Objectives

* The main objective in terms of outcome was to create a framework for users which can help them suggesting the right songs for them.
* This project is to reduce the time that user generally wastes on looking for the right song.

## 

## Chapter 3

**Tools and Techniques**

## 3.1 System Architecture

* Windows- 64-bit x86, 32-bit x86
* MacOS- 64-bit x86
* Linux- 64-bit x86, 64-bit Power8/Power9.
* Minimum 5 GB disk space to download and install.

## 3.2 Operating System Requirement

Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.

If your operating system is older than what is currently supported, you can find older versions of the Anaconda installers in our [archive](https://repo.anaconda.com/archive/) that might work for you. See [Using Anaconda on older operating systems](https://docs.anaconda.com/anaconda/install/#old-os) for version recommendations.

**3.3 Software and Library Used**

## 3.3.1 Python Programming Language

Python is a high level, interpreted, general purpose programming language, created by Guido van Rossum and first released in 1991. [1] Python supports multiple programming paradigms such as (particularly, procedural), object-oriented, and functional programming. For the first time python was conceived in the 1980s and later python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting. Python 3.0, released in 2008 was a major revision where most of the python 2 code does not run unmodified on python 3.

## 

## 3.3.2 NumPy and Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides an in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels.

Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

**“Import pandas as pd”**

Data frames can also be easily exported and imported from CSV, Excel, JSON, HTML and SQL database. Some other essential methods that are present in data frames are:

* **head ():** returns the top 5 rows in the data frame object
* **tail ():** returns the bottom 5 rows in the data frame
* **info ():** prints the summary of the data frame
* **describe ():** gives a nice overview of the main aggregated values over each column

## 3.3.3 Matplotlib

Matplotlib is a 2d plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments. Matplotlib can be used in Python scripts, Python and IPython shell, Jupyter Notebook, web application servers and GUI toolkits. The matplotlib Python library, developed by John Hunter and many other contributors, is used to create high-quality graphs, charts, and figures. The library is extensive and capable of changing very minute details of a figure.

## 3.3.4 Scikit-learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Extensions or modules for SciPy are conventionally named [Scikit](http://scikits.appspot.com/scikits). As such, the module provides learning algorithms and is named scikit-learn.

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as ease of use, code quality, collaboration, documentation and performance. Although the interface is Python, c-libraries are leveraged for performance such as NumPy for arrays and matrix operations, [LAPACK](http://www.netlib.org/lapack/), [LibSVM](http://www.csie.ntu.edu.tw/~cjlin/libsvm/) and the careful use of Python.

**3.3.4.1 TF-IDF**

TF-IDF stands for term frequency-inverse document frequency. It highlights a specific issue which might not be too frequent in our corpus but holds great importance. The TF–IFD value increases proportionally to the number of times a word appears in the document and decreases with the number of documents in the corpus that contain the word. It is composed of 2 sub-parts, which are:

* Term Frequency (TF)
* Inverse Document Frequency (IDF)

**3.3.5 Seaborn**

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

## 3.3.6 Jupyter Notebook

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter](http://jupyter.org/). Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use. The next most popular distribution of Python is [Anaconda](https://www.anaconda.com/). Anaconda has its own installer tool called Anaconda that you could use for installing a third-party package. However, Anaconda comes with many scientific libraries preinstalled, including the Jupyter Notebook, so you don’t actually need to do anything other than install Anaconda itself.

# 

# Chapter 4

# Methodology

# As mentioned above music recommendation engine are of various types and using different algorithms we can get different results even if we use different approach to do same algorithm it gives different results. As shown in the given below diagram firstly we input data after analyzing the data we try to extract the features. After all this we try to implement the algorithms and then result. Now let’s discuss one by one.

# 

# Fig 4.1: -Basic steps for Music Recommendation System

## 4.1 Dataset

## We have used Kaggle.com for a dataset where “Spotify dataset (1922-2021)” [1] dataset is downloaded. This dataset data.csv and data\_w\_genres.csv files in which acousticness, artists, danceability, duration, energy, id, instrumentalness, key, liveness, loudness, mode, name, release date, speechiness, tempo, valence. Each column of the dataset carries 174K data. We are going to predict the song on the bases of popularity name and genres features.

## 4.2 Data Cleaning

## This dataset needed some cleanings and modification. Besides, some feature representation should be done. At the initial stage we have two datasets data.csv, data\_w\_generes.

* + - We convert genres from string to list.
    - Now dropping all columns from data\_w\_generes except id, generes\_udp (in this coulumwe put generes all together).

## Dropping duplicate dataset

## Dropping data having 0 popularity.

## Reducing dataset from 158k to 4k because we can’t process this big data.

## We create rating according to popularity by dividing by10 so that we can get more accurate data.

## 

## Fig 4.2: - Graph show popularity of songs in data after losing data

## 4.3 Feature Extraction

## We use TF-IDF (Term Frequency Inverse Document Frequency) for genres in Content-based recommendation engine so that we can convert each word to feature index in the matrix so that each token gets a feature index. **Term Frequency (TF)** gives us the frequency of the word in each document in the corpus. It is the ratio of number of times the word appears in a document compared to the total number of words in that document. It increases as the number of occurrences of that word within the document increases. Each document has its own tf. **Inverse Data Frequency (IDF)** used to calculate the weight of rare words across all documents in the corpus. The words that occur rarely in the corpus have a high IDF score. It is given by the equation below.

## In Collaborative filtering we extract popularity of song from database but there is problem. In music peoples don’t give ratings or any other thing so we consider no. of times the song is listened is popularity. Considering popularity, the main pillar for collaborative filtering we extract the features.

## 4.4 Algorithms Used

## KNN is a distance-based classifier, meaning that it implicitly assumes that the smaller the distance between two points, the more similar they are. In KNN, each column acts as a dimension. In a dataset with two columns, we can easily visualize this by treating values for one column as X coordinates and the other as Y coordinates. Since this is a Supervised learning algorithm*,* we must also have the labels for each point in the dataset, or else we wouldn’t know what to predict. We have used three different KNN technique Cosine Similarity, Euclidean distance and Manhattan distance.

## 4.4.1 Cosine Similarity

## In this project we have used KNN cosine similarity. The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, it can be seen as a comparison between genres on a normalized space because we’re not taking into the consideration only the magnitude of each word count (tf-idf) of each line, but the angle between the documents. What we have to do to build the cosine similarity equation is to solve the equation of the dot product for the \cos{\theta}:

## 

## Fig 4.4.1.1: - Cosine similarity formula [14]

And that is it, this is the cosine similarity formula. Cosine Similarity will generate a metric that says how related are two documents by looking at the angle instead of magnitude, like in the examples below:

## 

## Fig 4.4.1.2: - Cosine similarity graph [15]

## 4.4.2 Euclidean Distance

## Euclidean distance is one of the most used distance metrics. It is calculated using Minkowski Distance formula by setting p’s value to 2*.* This will update the distance ‘d’ formula as below.

## 

## Fig 4.4.2.1: - Euclidean Distance Formula [16]

## For each dimension, we subtract one point’s value from the others to get the length of that “side” of the triangle in that dimension, square it, and add it to our running total. The square root of that running total is our Euclidean distance. Just as with Manhattan distance, we can generalize this equation to n dimensions.

## 4.4.3 Manhattan distance

## We use Manhattan Distance if we need to calculate the distance between two data points in a grid like path. We use Minkowski distance formula to find Manhattan distance by setting p’s value as 1. The easiest way to remember Manhattan distance is to use the analogy that provides this distance metric. Formula given below.

## 

## Fig 4.4.3.1: - Manhattan Distance Formula [16]

## In this project we use all these algorithms in order to give recommendations.

## 

## 4.5 Result

## To find the recommendation for the song we have to search songs. Different algorithms show different recommended songs. Here are few recommended songs.

## 

## Fig 4.5.1: - Result On the bases of Popularity (Cosine Similarity)

## In the figure 4.5.1 Collaborative filtering result is shown. In this recommendation is based on popularity using KNN cosine similarity algorithm.

## 

## Fig 4.5.2: - Result On the bases of Popularity (Euclidean distance)

## In the figure 4.5.2 Collaborative filtering result is shown. In this recommendation is based on popularity using KNN Euclidean distance.

## 

## Fig 4.5.3: - Result On the bases of Popularity (Manhattan distance)

## In the figure 4.5.3 Collaborative filtering result is shown. In this recommendation is based on popularity using KNN Manhattan distance.

# 

## Fig 4.5.2: - Results On the bases of Genres (content-based filtering)

## In the figure 4.5.3 2 Content-based filtering result is shown. In this recommendation is based on genres as we can see in figure popularity have variation. These songs are recommended using KNN cosine Similarity algorithm.

# Chapter 5

# CONCLUSION AND FUTURE SCOPE

# Conclusion

## This project work has provided us with a chance to understand how recommendation systems work in large companies by getting a hands-on experience implementing our own system. We realized that song recommendation is a tough problem which yields low precision values both in our work and all the other related works we studied. We implemented and evaluated song recommendation systems using Content Based Filtering and Collaborative Filtering. From what we understood, the collaborative filtering method proves to be easier to implement and evaluate, as it runs primarily on a user’s choices, as opposed to the inherent features within a song, which may or may not represent the user’s prediction towards the songs. KNN cosine similarity gives good recommendation on the bases of genres and popularity.

## Future Scope

## We can create a platform for commercial purpose using Spotify API and apple API. With recommendation we can provide editing, creating song list.

As mentioned in Part II related work, there are a lot

of key variables affect house prices. If data are available,

a good idea is to introduce more features, for example

income, salary, population, local amenities, cost of living,

annual property tax, school, crime, marketing data.

Furthermore, Random forest is an advanced

regression algorithm; it may help to improve prediction

accuracy.

Finally, we suggest building a separate algorithm to

detect and predict abnormal transactions SalePrice

## REFERENCES

## [1] https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks?select=data.csv

## [2] https://expertsystem.com/machine-learning-definition/

## [3]https://www.datarobot.com/wiki/prediction/#:~:text=What%20does%20Prediction%20mean%20in,will%20churn%20in%2030%20days.

## [4]https://www.researchgate.net/publication/339172772\_Recommender\_Systems\_An\_Overview\_Research\_Trends\_and\_Future\_Directions

[5] https://buffer.com/library/social-media-sites/

[6] https://towardsdatascience.com/recommender-systems-the-most-valuable-application- of-machine-learning-part-1-f96ecbc4b7f5

[7] https://www.marketsandmarkets.com/Market-Reports/recommendation-engine- market-151385035.html

[8] https://colab.research.google.com/noteboo

[9] https://www.bluepiit.com/blog/classifying-recommender-systems/

[10] https://www.sciencedirect.com/topics/computer-science/cosine-similarity

[11] https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed

[12] https://www.sciencedirect.com/science/article/pii/S1110866515000341

[13] https://medium.datadriveninvestor.com/product-recommendation-based-on-visual-similarity-on-the-web-machine-learning-project-end-to-end-6d38d68d414f

[14] https://dataaspirant.com/five-most-popular-similarity-measures-implementation-in-python/

[15] https://deepai.org/machine-learning-glossary-and-terms/cosine-similarity

[16] https://medium.com/@luigi.fiori.lf0303/distance-metrics-and-k-nearest-neighbor-knn-1b840969c0f4