

# ASSIGNMENT 5

Divyanshi Kamra, Naman Agrawal, Tushar Nandy, Prasann  
Viswanathan

June 2, 2020

## Abstract

This was by far the most fun project and we've probably spent maximum time on listening to most of the songs in our data-base, broadening our musical horizons. Jokes aside here's our journey in a chronological format.

## 1 23rd April (Morning)

Since Sanjev has made it clear not to spare any detail from our thought process, we started off by the thing that was most obvious to us and didn't require much thought; Classification of our data. We created a shareable spreadsheet, entering all our songs and proceeding on very dull and basic lines, we added just the attributes mentioned compulsory by Sanjev; Happy, Sad and Romance. We further realized that Rap God can't really be classified as any of those hence we decided to add another column; Fast to signify Rap or exercise music. Our model seemed basic and prospects of a success looked glum.

We split the work here and filled in entries for Happy, Sad, Romance and Fast-ness rating them all out of 10. (Hilarious bloopers alert).

We halfheartedly decided that the song with closest resemblance to the previous song should be the recommended one and Tushar prepared a basic code for the same, referring to a link shared and learnt by all of us. [1]

## 2 23rd April (The Video Conference)

Boy, was this a wake up call. I kid you not when i say we were dismally under-prepared for the grilling by Sanju. The truth was we hadn't given enough time to the project yet and we were suffering. Sanju brought forward a lot of important points and a few ideas for implementation that we listened to earnestly. Then it was time to grind.

We realized that this problem was fairly simple. The coding wouldn't be a major issue as pointed out by Divyanshi as both Divyanshi and Tushar are self-proclaimed code-blooded. (I can't say the same for myself xD).

The problem could be split into the issue of adding important attributes that influence the recommendations followed by implementing a feedback mechanism which tweaks the values of skipped songs, hence making them less often recommended.

## 3 24th April(Early morning/Late Night?)

As our early bird Tushar peacefully drifted to dreamland, we decided to perfect our model as far as possible. On research from various sources online (Which involved me googling "How does song recommendation work?" in different ways) i came across the following conclusions; Much more than mood of the person, the popularity of the song influences it's play time. Basically, a mainstream song is everyone's guilty pleasure. Hence the first change we needed to make; find a way to involve popularity. Solution-Youtube Views.[2] Next, were the moods that we were considering absolutely reliable? Did they ensure the entire emotional spectrum was taken into account of? The answer was kinda yes. As advised, we decided to do away with both happy and sad instead defining Happy-Sad. Positive values signalling Happy songs and negative, sad. Apart from this we kept romance and speed (slow to fast) as the other mood influencing attributes. A myriad of websites described music by at-least 20 odd moods including dramatic, energetic, heartbroken etc but our logic argued that who has the time to accurately define whether they are being ad or downright dramatic?! Hence we kept the moods simple to define. [3] Next thing to account for, genre. Well this is debatable. Only a true music aficionado would know accurately which genre a song belongs to, however this attribute was necessary as there were quite a few rap, EDM and rock songs in our data base, hence implying that some people would prefer (or

hate) certain genre's passionately. Added genres and kept them simple. Pop has so many sub-categories but we stuck with pop classic and pop rock. (As none of us have a PHD in music theory).

[4] Last but not least, artist influence on music. You just heard a song by eminem, you'd probably be annoyed to hear a JB song next; even if its equally popular and surprisingly matches your mood. We also have assumed that the artist influences as much as the genre does. Hence providing them the same scaled factor.

Please note that myself and Naman are not slackers xD. We did the grueling task of entering the values in our data base while Tushar and Divyanshi copy pasted code snippets from websites ( ° ° )

Last time I checked data entry was a paid job xD. (JK Tushar Divyanshi, don't kill me)[5]

## 4 24th April Morning

As clearly mentioned by Sanju, he needed a basic UI for the user. Hence a basic skeleton of that was created. Now we has to fill in the major funtions and implement KNN on the songs keeping in mind all of their attributes and scaling values down if the song is skipped. Here's our UI skeleton;

[6]

```
# Assuming that all the relevant information is in lists or numpy arrays

import numpy as np

print(''WE1COME...!!!'')

song_current = input("\n Please type your first song: ")

while True:

    # Code for knn selection

    song1, song2, song3 = ['A', 'B', 'C']
    print(f''\n Currently playing: {song_current}
```

```

        Upcoming songs: 1. {song1}  2. {song2}  3. {song3}\n'''

command = input("Type command here: ").lower()

if command == "next":
    song_current = song1

    # some code for scoring the recommendation

elif command == "skip-next":
    song_current = song2

    # some code for scoring, different as compared to the previous one

elif command == "search":
    song_current = input("What would you like to hear? ").lower()

    # code for negative marking

elif command == "exit":
    break

```

We've included the code for the UI as it's imperative that the reader knows at-least on a face level what we wish to accomplish with our music recommend-er.

## 5 24th April-Noon to Evening

From here began giving our skeletal UI some meat and muscle. Giving it the ability to move and do stuff.

Implementing KNN was easy, [7] we decided to base our next recommendation on the previous 4 songs heard by the user. Hence maintaining the flexibility in mood (which changes every 20 minutes) but maintaining some stability and past influence as well.

Each attribute contributed to a direction in a N-dimensional plane. The song nearest to average of 4 previous song vectors is recommended. (Granted that the recommendations aren't ideal initially; However this is because the values that WE have defined for each song may not be ideal (Tushar probably

plays Astronomia on his first dates).

As suggested by our mentor, the algorithm used to make changes to a skipped song is exponential. This is because if one or two people dislike a recommendation, it doesn't necessarily imply the recommendation is wrong; However if many do, you've probably classified Faith as a happy song.

As mentioned in our previous section, each of next, skip=next and search produce a different change in the database. We have taken into account how each does 'by feel', but tweaking of the values should give an idealized music recommend-er.

At last we have enabled a accuracy rating sort of feed back system to check how many times did we recommend a song that the user agreed with. Probably with a large enough user-base the model may reach 100percent accuracy. The code has been provided here [8].

## 6 24th April-Now

Our model runs well! (I'm as surprised as you are). It does exactly what the previous section said it would do.

As Sanju has bluntly called me out for not contributing anything, I'd like to take a few lines to defend myself here.

I read and understand all of the coding that goes on here. All of it. Tushar and Divyanshi were well on their way to completing all the functions and I know my place in the team is the one who faces the brunt of our mentor as he grills us on our documentation (always prepared by me, I wonder why?) Now that my I have justified myself. Here are a few runs of our machine you can feast your eyes upon;

```

WELCOME...!!!

Please type your first song: Memories
Currently playing: Memories
Upcoming song:Hotline Bling, Star Boy

Type command here: next
Currently playing: Hotline Bling
Upcoming song:Star Boy, Photograph

Type command here: next
Currently playing: Star Boy
Upcoming song:Photograph, Faith

Type command here: next
Currently playing: Photograph
Upcoming song:Faith, We Don't Talk Anymore

Type command here: skip-next
Warning (from warnings module):
  File "C:\Users\kamra\Desktop\Ass5\team-1a\running_project.py", line 50
    df[mood][song_num] -= 0.5
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

Currently playing: We Don't Talk Anymore
Upcoming song:Memories, Thinking Out Loud

Type command here: exit
Accuracy of our predictions in this run:
75.0

```

Figure 1: Run 1

```

WELCOME...!!!

Please type your first song: Taki Taki
Currently playing: Taki Taki
Upcoming song:Havana, 2U

Type command here: next
Currently playing: Havana
Upcoming song:2U, Steal My Girl

Type command here: next
Currently playing: 2U
Upcoming song:Steal My Girl, Baby

Type command here: next
Currently playing: Steal My Girl
Upcoming song:Baby, What Do You Mean

Type command here: search
Choose the song you wish to play:
Squeezed text (61 lines).
What would you like to hear? Electricity

Warning (from warnings module):
  File "C:\Users\kamra\Desktop\Ass5\team-1a\running_project.py", line 50
    df[mood][song_num] -= 0.5
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

Currently playing: Electricity
Upcoming song:Taki Taki, What Do You Mean

Type command here: next
Currently playing: Taki Taki
Upcoming song:What Do You Mean, Baby

Type command here: exit
Accuracy of our predictions in this run:
60.0

```

Figure 2: Run 2

```

C:\Users\kakra\Desktop>python running_project.py
WELCOME.....!!

Please type your first song: Memories
Currently playing: Memories
Upcoming song:Hotline Bling, Star Boy

Type command here: next
Currently playing: Hotline Bling
Upcoming song:Star Boy, Photograph

Type command here: next
Currently playing: Star Boy
Upcoming song:Photograph, Faith

Type command here: next
Currently playing: Photograph
Upcoming song:Faitth, We Don't Talk Anymore

Type command here: skip-next
Warning (from warnings module):
  File "C:\Users\kakra\Desktop\Ass5\team-1a\running_project.py", line 50
    df[mood][song_num] -= 0.5
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
Currently playing: We Don't Talk Anymore
Upcoming song:Memories, Thinking Out Loud

Type command here: exit
Accuracy of our predictions in this run:
75.0

```

Figure 3: Run 3

Please note that the red message is just a warning that changes have been made to the csv file after you've skipped a particular song. (Sorry for the lack of clarity in the images, the image files shall be uploaded on github as well)

This is important:

If you wish to run the program yourself and test out our music recommend-er, do the following;

1. Download the csv file as well as the running project file from the link.
2. Save them both in THE SAME FOLDER
3. Run the program
4. Each instruction is case sensitive and so are the songs, so in your first song selection, make sure to enter the song AS IT IS in the csv file.

## References

Our first attempt at the code

[https://github.com/divyanshikamra/team-1a/blob/master/Ass5\\_first\\_attachment.ipynb](https://github.com/divyanshikamra/team-1a/blob/master/Ass5_first_attachment.ipynb)

The reference that suggested we should include popularity

<https://towardsdatascience.com/how-to-build-a-simple-song-recommender-296fcbc8c85>

The reference that suggested we should include moods

<file:///C:/Users/HOME/Downloads/applsci-08-01103.pdf>

The reference that suggested we should include genres

<https://www.connollymusic.com/stringovation/different-genres-of-music-affect-your->

KNN algorithm used

<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbor>

UI skeletal code

<https://github.com/divyanshikamra/team-1a/blob/master/UI.py>

A fun video on KNN

<https://youtu.be/6kZ-OPLNcgE>

The final running code

[https://github.com/divyanshikamra/team-1a/blob/master/running\\_project.py](https://github.com/divyanshikamra/team-1a/blob/master/running_project.py)