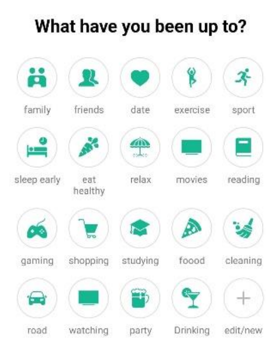
CS 210 COURSE PROJECT Cem Dereli 30708

Personalized Track Recommendations Based on Mood and Activity Using Machine Learning

Mental health is an important aspect of overall human health that is often being overlooked. It affects almost every part of human life and decreases quality of life if it's not treated. Understanding behaviors and learning how to break mental health decreasing cycles is an important part of dealing with mood related problems. I keep track of my mood and my daily habits to reflect and learn about myself in order to grow and improve my mental health.

As a music enthusiast, I listen to music almost every day and my music listening habits are shaped by my life, personality, and lifestyle. My mood is one of the biggest factors that is affecting my daily life and thus shapes the songs I listen to. I don't just use Spotify to listen to music, I also use it to meditate, relax and sleep. My daily activities are hand to hand with the playlists I choose and music I listen to. This project is aiming to recommend songs to me based on my mood and daily activity. By completing this project, I won't just learn a lot about my mental health and listening habits, I will also be able to pick the track that suits most to my mood and activity of the day.

The data of the project comes from two different sources. The first source is Spotify, and it is my extensive streaming history. The second source is a mood tracker and digital journaling app called Daylio Journal – Mood Tracker. I have been using that app since the first semester of the freshmen year. The collection of data was done by requesting and downloading the data from both apps. The Daylio app collects data regarding my mood and daily activities by simply sending a notification to my phone every day.

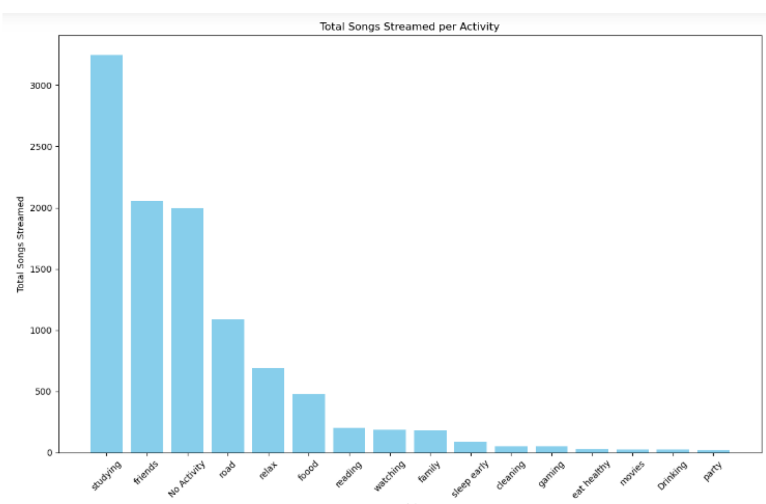


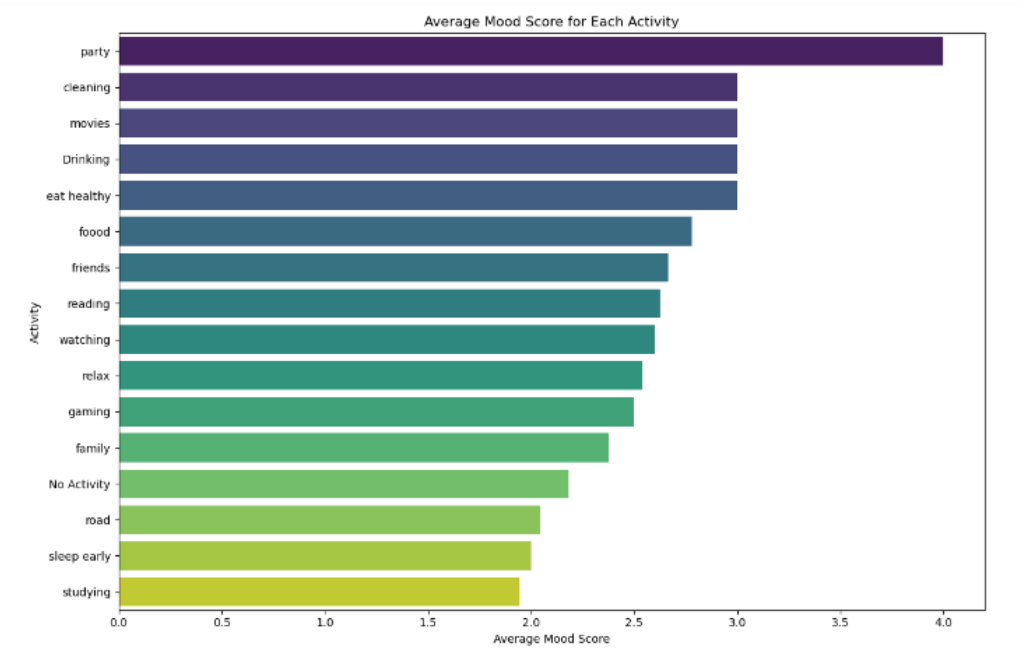
Here is the mood range from "rad" being the best to "awful" being the worst mood. That app also collects data regarding the daily activities and then helps you to track your mood and habits. I scraped both data from the files after requesting and downloading from the apps.

	full_date	date	weekday	time	mood	activities	note
0	2022-03-15	March 15	Tuesday	21:48	good	relax | studying	Neutral
1	2022-03-16	March 16	Wednesday	23:47	good	friends | relax | reading	Neutral
2	2022-03-17	March 17	Thursday	21:18	good	friends | foood	Neutral
3	2022-03-18	March 18	Friday	20:00	meh	family | relax | road	Neutral
4	2022-03-19	March 19	Saturday	00:16	good	reading | watching	Neutral

The data is downloaded as CSV and JSON files and the extraction of the data is handled easily. Unnecessary columns such as column “note_title” is dropped from the mood data. After that missing values are handled: missing values in the 'note' and 'activities' columns are filled with 'Neutral' and 'No Activity' respectively. The “note” column is the journal part of the Daylio app and I don’t want to share sensitive information. Standardization is applied to the “note” column and any value that is not 'On track', 'Relapsing', or 'Unstable' is replaced with 'Neutral'. The mood data is reversed to align with the streaming data and in both datasets, date columns are converted to datetime objects. This was needed to merge two datasets on the date column. Before that, both datasets are filtered to have matching dates. Some inconsistencies are handled, and the data was ready for visualization, analysis, and training. The final version of the mood data frame can be seen in the figure above.

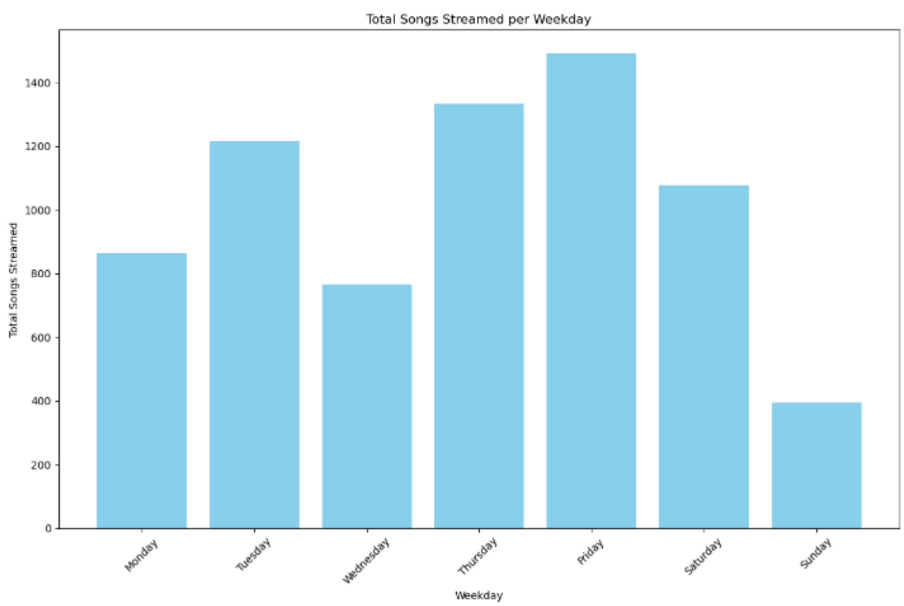
After that, relationships between activities, streaming habits, mood scores and days of the week were analyzed. The relationship of mood scores and streaming habits with activities are the visualized like the following:



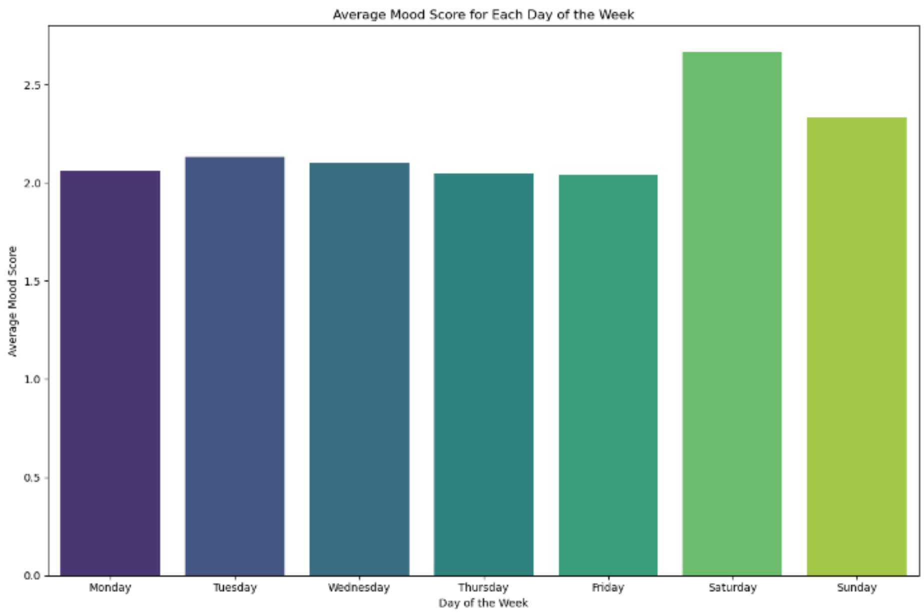


Looking at the first figure, it can be seen that, during some activities, I tend to listen to way more music. And interestingly, my average mood score is relatively low in some of these activities. For example, I am listening to the music most when I am studying but my mood score when I study is the lowest. Activities such as "road" and "no activity" tend to be lower in mood scores but I stream a lot of tracks when my daily activities are these activities. Other observations can be seen such that I feel happier when I party or clean and stream less tracks during these activities.

Other visualizations regarding relationships are "Average Mood Score for Each Day of the Week" and "Total Songs Streamed per Weekday".



This graph shows when I tend to listen to music and "Sunday" is undeniably lower.



The mood score is relatively low on Mondays and high on Saturdays.

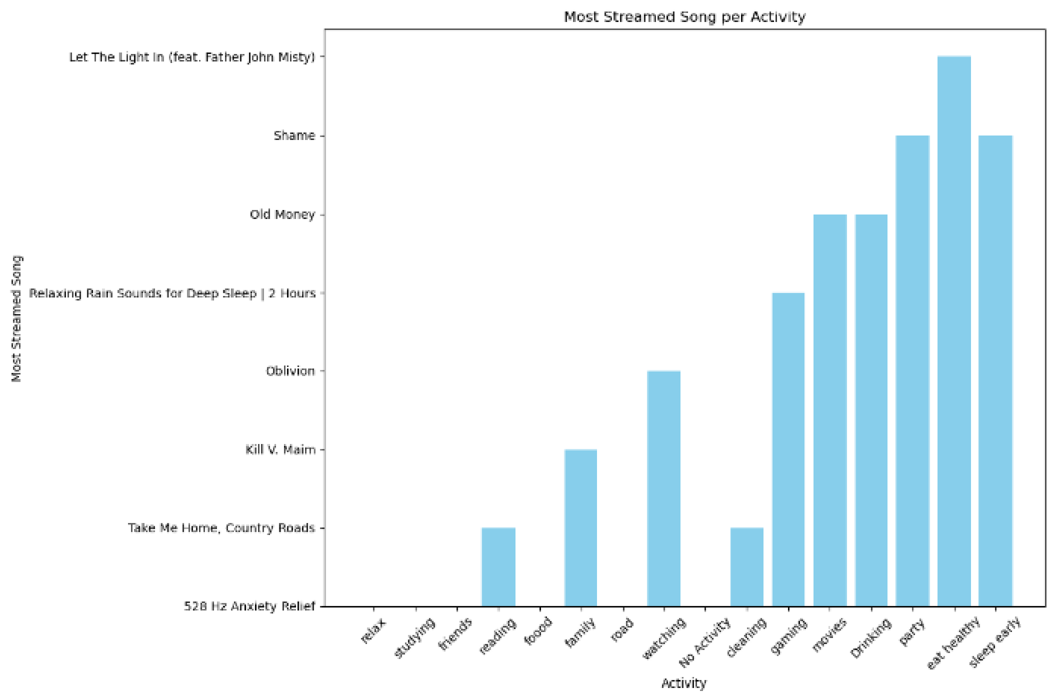
Another important observation that observed is the most listening track on each day. This was the most surprising (but also not surprising) part of the observations:

	weekday	master_metadata_track_name	play_count
7	Friday	528 Hz Anxiety Relief	92
593	Monday	528 Hz Anxiety Relief	243
890	Saturday	528 Hz Anxiety Relief	214
1114	Sunday	528 Hz Anxiety Relief	55
1249	Thursday	528 Hz Anxiety Relief	107
1670	Tuesday	528 Hz Anxiety Relief	247
1932	Wednesday	528 Hz Anxiety Relief	191

(Anxiety Relief Frequencies)

This is the clearest evidence that underlies the fact that my listening habits are being shaped by my mood and mental state since I use Spotify to meditate, calm down and focus. It is also worth mentioning that even though I stream relatively less on Mondays than Saturdays, I still manage to listen to Anxiety Relief Frequencies more on Mondays (243 >214). It also aligns with my mood scores since my mood score on average is way on Saturdays compared to Mondays.

In order to understand the data better before moving to machine learning and training process, I wanted to see the relationships between songs and activities. The following visualization shows which song I listen to the most when I am doing a certain activity.



This was the last visualization in this project to explore and understand the data and analyze relationships.

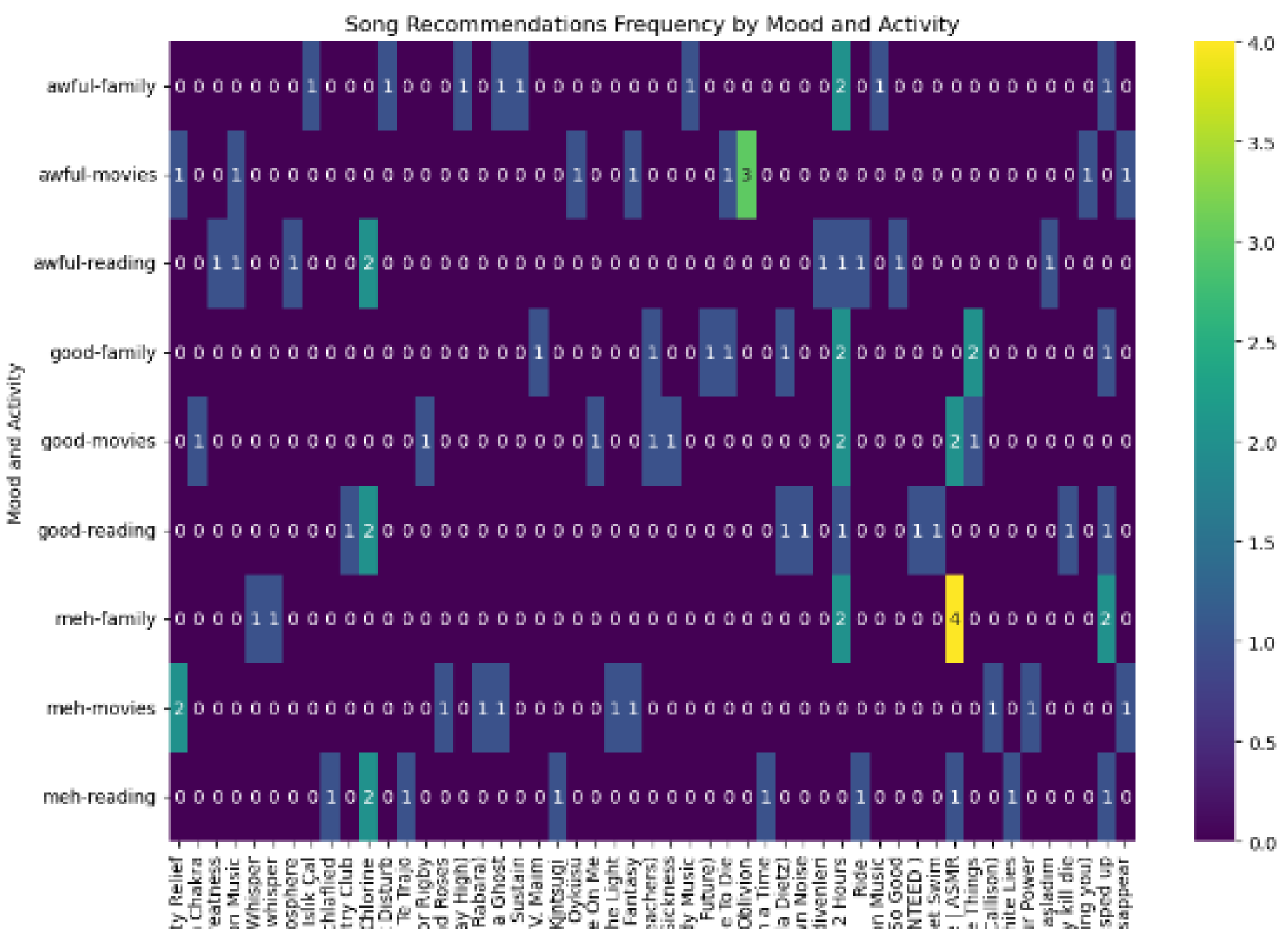
For the next part I have come up with two different machine learning algorithms to recommend a song based on activity and mood. This part was the most complicated part of the project and faced some limitations which will be evaluated later in this report. The first machine learning algorithm is a straightforward machine learning model which uses “Random Forest Classifier”. Its aim is to recommend a song based on mood and activities. Before the training part, the moods are **encoded** awful being the 0 and the rad being the 4, which can be done because there is a ranking in moods. Since activities are categories and can’t be encoded the way moods are encoded, **one-hot encoding** method is used to encode them. The song names are spitted as **target(y)** and encoded moods and activities become **features(x**). The random forest classifier is trained on the feature set to predict the most likely song. However, this first method relies only on the model’s output probabilities and some tracks that are listened more can dominate the recommendations. Some example outputs and heatmap visualization can be seen as:

```
# Test the function
example_mood = 'bad'
example_activities = ['studying']
recommended_song = predict_song(example_mood, example_activities)
print(f"Recommended track for mood '{example_mood}' and activities '{
```

```
Recommended track for mood 'bad' and activities '['studying']': Norm
an fucking Rockwell
```

```python
# Test the function
example_mood = 'rad'
example_activities = ['friends']
recommended_song = predict_song(example_mood, example_activities)
print(f"Recommended track for mood '{example_mood}' and activities '{
```

```
Recommended track for mood 'rad' and activities '['friends']': Other
Friends (feat. Sarah Stiles, Zach Callison, Deedee Magno Hall, Estel
le & Michaela Dietz)
```



The frequency of the songs that are being recommended by the first algorithm based on the activity and the mood.

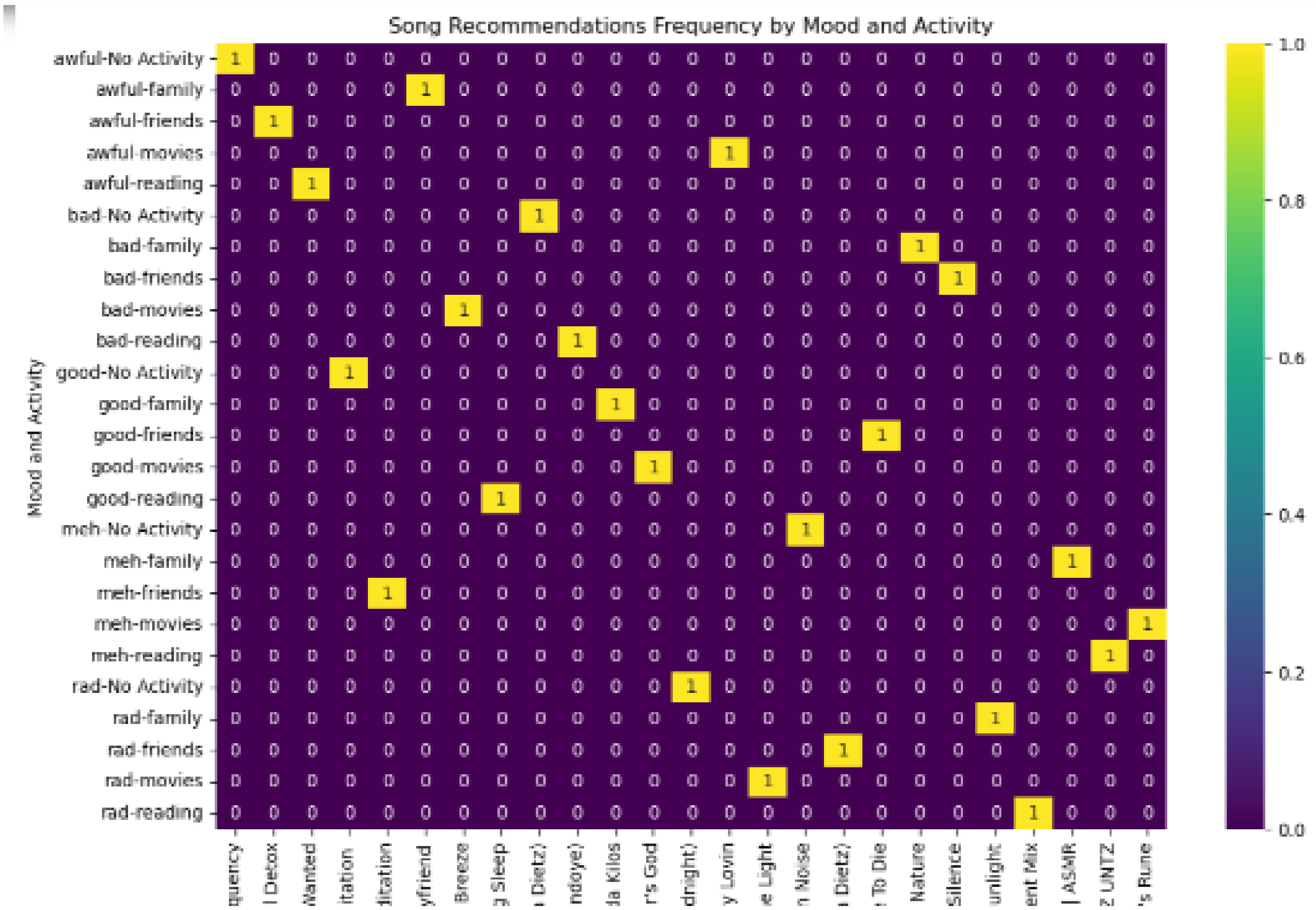
The second algorithm is way better at recommending songs because it calculates association scores that represent the frequency of a song being associated with particular mood and set of activities. Output probabilities are adjusted based on the association scores and then these adjusted probabilities are normalized. This second machine learning method provides an enhanced personalization and more accurate recommendations. Some outputs and heatmap visualization of this model are:

```python
# Test the function
example_mood = 'awful'
example_activities = ['studying']
recommended_song = predict_song(example_mood, example_activities)
print(f"Recommended track for mood '{example_mood}' and activities '{
```

```
Recommended track for mood 'awful' and activities '['studying']': Bi
polarsan Islık Çal
```

```python
# Test the function
example_mood = 'good'
example_activities = ['friends']
recommended_song = predict_song(example_mood, example_activities)
print(f"Recommended track for mood '{example_mood}' and activities '{
```

```
Recommended track for mood 'good' and activities '['friends']': all-
american bitch
```



The X axis here are the songs and the Y axis are the combinations. The big setback of this model is there is too much computation in the process.

Other big setbacks are the problem that there are not enough activity – mood combinations and some classes have too little data that the model can't really predict well for some combinations.