Capstone: Music Recommendation System

Western Governors University

Jacob Boyd

Student ID: 001393514

Table of Contents

[Prompt A 4](#_Toc101367150)

[Letter of Transmittal 4](#_Toc101367151)

[Project Recommendation 6](#_Toc101367152)

[Problem Summary 6](#_Toc101367153)

[Application Benefits 6](#_Toc101367154)

[Application Description 7](#_Toc101367155)

[Data Description 7](#_Toc101367156)

[Objective and Hypothesis 8](#_Toc101367157)

[Methodology 8](#_Toc101367158)

[Funding Requirements 9](#_Toc101367159)

[Stakeholders Impact 9](#_Toc101367160)

[Data Precautions 9](#_Toc101367161)

[Developer Expertise 9](#_Toc101367162)

[Prompt B 10](#_Toc101367163)

[Project Proposal 10](#_Toc101367164)

[PROBLEM STATEMENT 10](#_Toc101367165)

[CUSTOMER SUMMARY 10](#_Toc101367166)

[EXISTING SYSTEM ANALYSIS 10](#_Toc101367167)

[DATA 11](#_Toc101367168)

[PROJECT METHODOLOGY 11](#_Toc101367169)

[PROJECT OUTCOMES 12](#_Toc101367170)

[IMPLEMENTATION PLAN 13](#_Toc101367171)

[EVALUATION PLAN 13](#_Toc101367172)

[RESOURCES AND COSTS 14](#_Toc101367173)

[PROGRAMMING ENVIRONMENT 14](#_Toc101367174)

[ENVIRONMENT COSTS 14](#_Toc101367175)

[TIMELINE AND MILESTONES 15](#_Toc101367176)

[Prompt C 16](#_Toc101367177)

[Post-implementation Report 16](#_Toc101367178)

[Project purpose 16](#_Toc101367179)

[Data product code 17](#_Toc101367180)

[Hypothesis verification 18](#_Toc101367181)

[Effective visualizations and reporting 18](#_Toc101367182)

[Accuracy analysis 19](#_Toc101367183)

[Application testing 19](#_Toc101367184)

[Appendices 20](#_Toc101367185)

[Application Files 20](#_Toc101367186)

[Installation Guide 20](#_Toc101367187)

[User Guide 21](#_Toc101367188)

[Summation of Learning Experience 21](#_Toc101367189)

[References 22](#_Toc101367190)

# Prompt A

## Letter of Transmittal

April 17th, 2022

Mr. Joey Mills

Spotted Shark Company

800 S Goggin Street

Brazil, IN 11111

Mr. Mills,

Due to a large and growing number of songs on the Spotted Shark database, customers have had a harder time finding new music they enjoy listening to. To solve this problem, the IT department recommends creating a system for recommending music that has similar attributes to music the client has expressed enjoyment in. The solution’s first version will allow users to input values ranging from zero to one via sliders that represent seven song/track attributes including: acousticness, danceability, energy, instrumentalness, liveness, speechiness, and valence. After the user submits the attribute values, the system will the recommend songs whose attributes are most like those desired by the user by displaying the songs name, the album it is on, and the artists who created the work. The system will be run locally on the user’s device requiring only connection to the existing spotted shark track database to populate the list of recommended songs. The song recommendation system will benefit the client in that it will lay the foundation for a more efficient and user-focused experience for finding new enjoyable music. Allowing users to find music they enjoy more easily will make the user more likely to continue to use the service, recommend the service to other users, and most-importantly enjoy the music they are listening to.

The objectives of the song recommendation system are to recommend songs to users both efficiently and accurately. The level of efficiency and accuracy in this project are tradeoffs. A solution that quickly recommends songs will be less accurate and a solution that takes much longer to recommend songs will be more accurate. The goal is to find an optimal area where on average the customer is not irritated by slow processing times and receives songs they will potentially enjoy listening to.

Due to the lack of new infrastructure that is required by this project and utilization of free developmental tools: building, installing, and maintaining this application is estimated to be $20,000 for the time and resources of the it staffs and likely another $4,000 per year of time and resources to maintain. The estimated costs include only the first version of the application and not future improvements.

To implement this solution, I will rely on the academic training culminated in a Bachelor of Computer Science with proficiency in validated through independent certifications from CompTIA, ITIL, and freeCodeCamp.

Sincerely,

Jacob Boyd

## Project Recommendation

### Problem Summary

The problem the song recommendation system will attempt to solve is that the large and growing number of songs in the Spotted Shark database is and will increasingly make the discovery of new songs that are enjoyable to the user more difficult. The solution will be to create a recommendation system that will allow users to discover music more easily. This solution will allow the user to input values that correspond to attributes of music such as acousticness, danceability, energy, instrumentalness, liveness, speechiness, and valence and will recommend songs based on the values representative of the preferences of the user. The first version of the project will recommend songs based on attribute similarity of tracks and will be created to be extensible for further versions of the applications; however, it will not initially extend beyond this scope into analysis of artists, albums, playlists, or user listening data such as skipping songs, etc.

### Application Benefits

Spotted Shark will benefit from this application for reasons that include but are not limited to the foundation of a more elaborate song recommendation system, analysis of its database of tracks, and allowing their users to be efficiently exposed to potentially enjoyable music. The application will also allow Spotted Shark to supplement their database with music that is more diverse in qualities to offer more variety for their listeners. This application will bridge the gap between the users and the vast growing ocean of songs in the Spotted Shark database.

### Application Description

The initial version of the application will be created for internal use and testing by Spotted Shark before it is available on mainstream browsers and operating systems that support the Spotted Shark parent application. The initial version of the application will be created using python 3.9.7 on jupyter notebook utilizing ipywidgets, matplotlib, spotipy, and sklearn. The application will employ data exploration techniques to analyze the Spotted Shark song database. It will also allow users to interact with sliders representative of song attribute values and a submit button which will print the song name, album, artists, and similarity score of the users input to the song attributes. The similarity score is the cosine similarity of the seven-dimension input by the user and the recommended track. A higher similarity score will represent a higher accuracy of recommendation.

### Data Description

The data used for this project will come in the form of a comma separated value file fitted to represent a table from a song database hosted by Spotted Shark. The csv file will be a subset of data found in a Spotify song data set found on Kaggle at the following link: <https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs>. The subset used for the project will have fourteen columns with the following attributes: song id, song name, album, album id, artists, artist ids, explicit, danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence. The data set will contain 1,048,576 rows of songs.

The pros of using this dataset are that the information is complete, containing no null values, and contains many songs with a large range of attributes allowing for more quality recommendations. The cons of the data set are that because it is large and will become larger over time it will be too time consuming to search for the most optimal recommendations. The application will therefore have to select recommendations from a smaller subset, so the output is displayed to the user in a timely manner.

### Objective and Hypothesis

The objective of the proposed project is to develop an extendable, scalable, and maintainable song recommendation system that allows users to efficiently find potentially enjoyable music that has attributes similar to the users desired attributes. The music recommendation system is premised on the theory that if the music has attributes that are more like the desired attributes input by the user, then the user will find the song more enjoyable, and the recommendation will therefore be more accurate.

### Methodology

The project will be managed using the industry standard methodology of Waterfall. The waterfall method which emphasizes documentation is important for the initial version of this project if it is to be improved upon by different teams or departments in the future. Also, the rigidity of the Waterfall method will help ensure that there is no scope creep and that the project will follow the projected timeline. The requirements stage of the project will detail the specific requirements of the application including the speed and average level of accuracy of recommendations desired. The design stage will be where the methods of data cleaning, analysis, and recommendation will be decided. The implementation stage will be where the development of the application takes place utilizing python and third-party libraries for data analysis and visualization. The verification stage will allow a beta program for Spotted Shark users to actively test the recommendation system and provide feedback on the speed and accuracy of the recommendations. The maintenance stage of the project will involve actively checking error logs and detailing bugs in Spotted Shark’s bug tracking system.

### Funding Requirements

Due to the lack of new infrastructure that is required by this project and utilization of free developmental tools such as Anaconda: building, installing, and maintaining this application is estimated to be $20,000 for the time and resources of the IT staff. To maintain the project another $4,000 per year of time and resources is projected. The estimated costs include only the first version of the application and not future improvements.

### Stakeholders Impact

The application will allow data analysis teams to better review the accuracy and efficiency of recommendations and visualize songs better via their attributes. The customers will therefore benefit from a more efficient and enjoyable user experience by allowing them to find music they enjoy easier than they could have beforehand.

### Data Precautions

The datasets used in this application are non-sensitive and therefore do not need protection.

### Developer Expertise

To implement the solution the spotted shark project team recommends a developer who has academic training culminating in Bachelor of Computer Science and has proficiency in the development and management of small software projects utilizing python libraries validated through independent certifications from CompTIA, ITIL, and other accredited certification organizations. In creating this application proficiency in machine learning algorithms, data analysis and data visualization will be necessary.

# Prompt B

## Project Proposal

### PROBLEM STATEMENT

The problem at hand is that due to the large and growing number of songs available in the Spotted Shark database users are unable to quickly find enjoyable music. The solution to this problem will entail the creation of an accurate and efficient recommendation system that allows users to easily view recommended songs based on attributes of the music they desire. This is an opportunity for Spotted Shark to create a more streamlined and enjoyable user experience.

### CUSTOMER SUMMARY

The users of this application will be users of the Spotted Shark parent application which is a music streaming service that allows users to listen to their favorite artists, songs, and create and share playlists. This project will be the foundation of the Spotted Shark music recommendation system and will be a jupyter notebook application. For the testing and analyzing data portion of the notebook, skills such as statistics and data analysis will be needed; however, the interactive model for recommending songs will be intuitive and user friendly, meaning it will require no training.

### EXISTING SYSTEM ANALYSIS

The existing environment is a database of songs and a parent music streaming application that is installed on third-party user devices. The project proposed will deliver a proof-of-concept application that will extend the Spotted Shark streaming service and allow users to easily find music with the specified desired attributes. New hardware infrastructure will be unnecessary for the initial version of the application.

### DATA

The data used for this project will come in the form of a comma separated value file fitted to represent a table from a song database hosted by Spotted Shark. The csv file will be a clean subset of data found in a Spotify song data set found on Kaggle at the following link: <https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs>. The subset used for the project will have fourteen columns with the following attributes: song id, song name, album, album id, artists, artist ids, explicit, danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence. The data set will contain 1,048,576 rows of songs.

The pros of using this dataset are that the information is complete, containing no null values, and contains many songs with a large range of attributes allowing for higher accuracy recommendations. The cons of the data set are that because it is large and will become larger over time it will be too resource and time intensive to search for the most optimal recommendations. The application will therefore have to select recommendations from a smaller subset, so the output is displayed to the user in a timely manner.

### PROJECT METHODOLOGY

The project will be managed using the industry standard methodology of Waterfall. The requirements stage of the project will detail the specific requirements of the application including the speed and average level of accuracy of recommendations desired. The design stage will be where the methods of data cleaning, analysis, and recommendation will be decided. The implementation stage will be where the development of the application takes place utilizing python and third-party libraries for data analysis and visualization. The verification stage will allow a beta program for Spotted Shark users to actively test the recommendation system and provide feedback on the speed and accuracy of the recommendations and ensure proper behavior of the application. The maintenance stage of the project will involve actively checking error logs and detailing bugs in Spotted Shark’s bug tracking system.

### PROJECT OUTCOMES

The following are a list of the in-process Project Deliverables:

* Requirements Document
* Software Architecture Design
* Documentation of Implementation Roadblocks
* Bug tracking documentation

The following are a list of the final Product Deliverables:

* Source Code for the Song recommendation system application
* Song database analysis notebook

**Implementation Plan** -- Explain how the project will be implemented. Your methodology material would work great here. Just reuse it. You may also include:

* Dependencies and milestones
* Deliverables – both tangible and intangible
* User testing -- if applicable

The following is the chronological plan for events to implement the music recommendation system:

1. Creation of Requirements document outlining the scope of the project
2. Obtain song data in csv format
3. Design of software architecture for project
4. Creation of source code for data analysis and music recommendation algorithm + Bug Tracking
5. Evaluation and verification of requirements utilizing user and in-house testing of application
6. Maintain documentation and source code for future related recommendation projects

### IMPLEMENTATION PLAN

The following is the chronological plan for events to implement the music recommendation system:

1. Creation of Requirements document outlining the scope of the project
2. Obtain song data in csv format
3. Design of software architecture for project
4. Creation of source code for data analysis and music recommendation algorithm + Bug Tracking
5. Evaluation and verification of requirements utilizing user and in-house testing of application
6. Maintain documentation and source code for future related recommendation projects

### EVALUATION PLAN

To evaluate and verify that the application meets the expectations set by the requirements document the user testing and feedback will be utilized. The accuracy of the song recommendations will be verified via two routes: average level of user satisfaction with recommendation and the cosine similarity value computed using the user desired attributes and potential song recommendations from the existing database.

### RESOURCES AND COSTS

Due to the lack of new infrastructure that is required by this project and utilization of free developmental tools: building, installing, and maintaining this application is estimated to be $20,000 for the time and resources of the it staff and likely another $4,000 per year of time and resources to maintain. The estimated costs include only the first version of the application and not future improvements.

### PROGRAMMING ENVIRONMENT

The project will require no new hardware infrastructure and utilize a python 3.9.7 jupyter notebook utilizing libraries such as pandas, ipywidgets, matplotlib, spotipy, and sklearn. The final data products and model for the initial version of this project will be displayed in the jupyter notebook format.

### ENVIRONMENT COSTS

Due to the utilization of free developmental tool Anaconda which includes VSCode, Jupyter notebook, and other data analytic tools, environment costs are projected to be minimal.

### TIMELINE AND MILESTONES

The main cost of developing the music recommendation application will be the time of human resources. The projected schedule is projected to require 60 hours of the course of 4 weeks.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Milestone | Pre-requisites | Activity | Resource Assigned | Hours | Start | End |
| 1 | - | Requirements Document | Project Manager | 10 | 3/1/22 | 3/4/22 |
| 2 | 1 | Architecture Design | Software Engineer | 10 | 3/7/22 | 3/11/22 |
| 3 | 1,2 | Develop Source Code for Application | Software Developer | 20 | 3/14/22 | 3/18/22 |
| 4 | 1,2 | Bug Tracking | Quality Assurance Personnel | 5 | 3/14/22 | - |
| 5 | 1,2,3,4 | Application Testing + Verification of Requirements from Milestone 1 | Users, Software Developer, and quality assurance officer | 10 | 3/21/22 | 3/25/22 |
| 6 | 1,2,3,4,5 | Project Lessons Learned | Project Manager, Software Engineer, Software Developer, and Quality Assurance officer | 5 | 3/28/22 | 3/31/22 |

# Prompt C

## Post-implementation Report

### Project purpose

The problem the application solved is to help users search the large and growing number of songs in the Spotted Shark database by developing the foundation of a Music recommendation system. The application allows users to find songs that closely match their desired attributes by inputting slider values respectively representing seven musical attributes including acousticness, danceability, energy, instrumentalness, liveness, speechiness, and valence.

The used to represent a song table from the spotted shark database is a comma separated value file which contains 1,048,576 which each have the following non-null valued columns: song id, song name, album, album id, artists, artist ids, explicit, danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence. The song id, song name, album, album id, artists, and artist ID columns are all string values, but the artists and artist ID column strings are in the form of lists. The Explicit column contains Boolean values corresponding to whether the song is considered offensive and/or unsuitable for children. The danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence columns all contain float values ranging from 0.0 to 1.0. The following is a list of what each value represents:

* Danceability: How suitable the song is for dancing
* Energy: Intensity and Activity
* Speechiness: How much of the song is speech-like
* Acousticness: Confidence whether the track is acoustic
* Instrumentalness: Level of vocal content in the song
* Liveness: Confidence whether the track was performed live based on audience sounds
* Valence: Positivity of track

### Data product code

The data analysis portion of the application explores the data set by utilizing pandas to read in data from the csv and displaying charts to the user using matplotlib plots. The notebook first displays a pie chart that shows the percentage of explicit vs non-explicit songs contained in the csv file. The next visual displays boxplots that allow the user to analyze the spread of each of the float value columns including danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence. The final data visualization is a hex bin plot in which the density of points from a danceability vs valence scatter plot can be analyzed.

The music recommendation portion of the application utilizes an interactive ipywidgets GUI based in the Jupyter notebook that allows the users to use slider values and a submit button to request recommended songs based on the danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence values. There is also a button for the user to reset the slider values, which sets each of the slider values to 0.5. When the submit button is clicked by the user the following events will take place:

* 1. The previous output in the GUI, if any, will be cleared
  2. Current slider values will be appended to a list representative of a seven-dimensional vector
  3. A subset of indices from the track\_features csv will be selected
  4. The main algorithm for recommendation will be called
  + For each songs located at the random indices collected before
    - A seven dimensional vector will be made from the float values of the danceability, energy, speechiness, acousticness, instrumentalness, liveness, and valence columns
    - The angle between the slider value vector and the track vector will be calculated using the cosine similarity method.
    - Each angle and index will be appended to a list of potentially recommended songs.
  + The list of potential songs will be sorted from most similar to least similar and returned

1. The Top 5 most similar tracks will be displayed in the GUI output

### Hypothesis verification

The initially established hypothesis was accepted based on the user testing of the speed and accuracy to which the application can recommend songs. The application was able to give the user similar music to the desired values input on the sliders in an acceptably efficient manner.

### Effective visualizations and reporting

The data was cleaned using an excel spreadsheet to select a subset of column values from the initial dataset. The initial data set did not contain any null values and data preparation beyond column selection and csv formatting was unnecessary.

The visualizations present in the application each help summarize the data and allow the user to analyze different aspects of the data by answering questions such as the following:

* 1. What percentage of the songs present are explicit vs child-friendly?
  2. What is the spread of each of the song attributes that the users will input in the GUI further in the application? Will there be a smaller pool for inputting more extreme values?
  3. What does the relationship between two of the attributes look like such as danceability and valence?

### Accuracy analysis

The application utilizes a cosine similarity value to represent the accuracy of a given recommendation to the user. The cosine similarity is used because it allows the system to calculate the angle between two vectors and theoretically how similar the attributes are to one another. Because the application utilizes a subset of the data to find the highest quality recommendations it is possible for the recommendations to be less accurate; however, the application was tested with varying sizes of subsets and was found to perform with acceptable levels of accuracy and efficiency when the number of songs considered for recommendation was 10^5.

### Application testing

Throughout the development of the application each segment of the application was tested for functionality by testing with differing portions of the dataset and testing each portion of the interactive GUI alone and paired with its associated functions.

Any errors that are detected by the application, a description will be input into the error log to help test, and make sure the application is maintainable in the future.

The application used usability and acceptance testing by users to find potential errors through use of the application and retrieve feedback from users about the efficiency and accuracy of the recommendations produced by the system.

# Appendices

## Application Files

This application only needs three main files to function as intended. The two files are the following:

* capstone\_notebook.ipynb
* track\_features.csv
* log.txt

## Installation Guide

Prerequisites:

Python 3.9.7, Pandas, Ipywidgets, Random, Matplotlib, Sklearn, Spotipy, Jupyter Notebooks

1. Install prerequisite applications
2. Extract the C964 file into the directory from which it will run
3. Open in Jupyter Notebooks on JN supported web browser

## User Guide

The application GUI will have a description of each of the sliders and labeled buttons that will allow for very user-friendly interaction.

## Summation of Learning Experience

Through building this application I was introduced to the large world of machine learning and large-scale data applications. I believe building this application has helped me learn more about researching for applications in the future as it forced me to look for efficient solutions to problems. I found many great sources that helped me understand more about machine learning and the limitless applications it has if the data is there, including this very application. The experience has contributed to my life-long learning in that it has helped me form a new way of looking at large scale problems. Standing on the shoulders of giants, I have been able to dip my toe in the ocean of possible applications of Machine Learning.

## References

This space intentionally left blank.