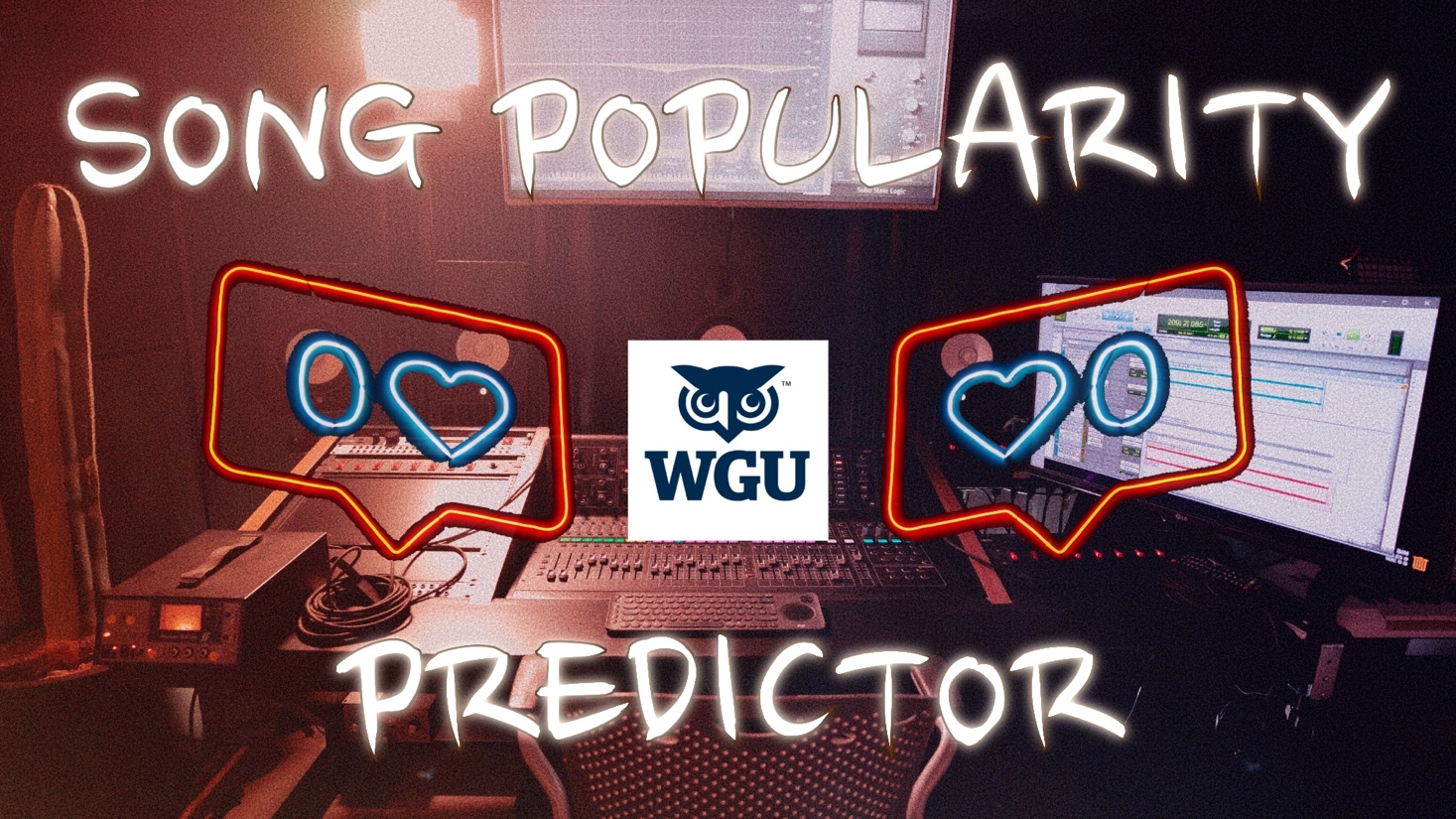
C964 – Computer Science Capstone

SONG POPULARITY PREDICTOR

Johannes Van Rossum

ID #001527666

WGU email: jvanros@wgu.edu

**10/15/2022**

# A1. letter of transmittal

Johannes Van Rossum

909 E Camelback Rd

Phoenix, AZ 85014

October 17, 2022

SONY Music Entertainment

25 Madison Ave

New York, NY 10010

Attn: Senior Leadership

MUSIC POPULARITY PREDICTOR – SOFTWARE PROPOSAL

Dear Senior Leadership,

“Too Many Songs, Not Enough Hits: Pop Music Is Struggling to Create New Stars” (Leight, 2022) is the eye-catching title of a recent article on Billboard.com. As you may well know from your own experience, building an audience for new acts is harder than ever due to an overflow of new music and songs in the marketplace. The same article goes on to say: “It’s a bigger and more level playing field, and everything is getting lost.” Over 80,000 tracks are being uploaded to major digital service providers each day and the songs that break seemingly do so by mere luck. Finding what is lost is now more time consuming, requires specialized personnel with unique skillsets, and is therefore more expensive. Machine Learning for Music & Media (MLMM), a company founded by a schooled musician, recording artist, and Machine Learning-engineer (with a Bachelor’s degree in Computer Science and independent certifications from CompTIA, ITIL, and EdX), dedicates itself to finding what got lost, to uncovering that gem from the muddied waters, based on the philosophy that true talent will always float to the surface. MLMM will develop for you a data-driven, web-based solution that will show it is not solely mere luck. The Song Popularity Predictor (SPP) is a tool to help clear the muddied waters and will predict for you a (new) song’s popularity based on its extracted musical attributes, such as danceability, energy, tempo, and much more. This application will benefit you in several ways:

* Unparalleled efficiency – sift through 80,000 songs in less than an hour and discover what new songs have the potential to be popular.
* Reduce costs – No more sitting through hours and hours of listening to bad demo tracks, no more wasting time on traveling to obscure nightclubs in all corners of the country, and no more need for a huge team of employees.
* Adaptability – adapting the model to comply with current music trends is easy and can be done at any time.
* Great potential for creative human & AI collaboration - Human perception is highly subjective and may be very different from what the AI thinks. As this field develops, humans can collaborate with AI to find new features that make a song popular, songwriters can use the tool to see whether they are on the right track with their new projects, and you and your artists can use SPP’s advice and predictions to enhance the product.

Development, implementation, and maintenance of this revolutionary tool will require an initial investment of $24,000 with a yearly maintenance cost estimated at 20% ($4,800).

Text

Description automatically generatedYours sincerely,

Johannes Van Rossum

# a2. Project Recommendation

## Problem Summary

With over 80,000 tracks being uploaded to major digital service providers each day the music industry market has become convoluted over the past years. Finding new talent and catchy songs has become more challenging, more time-consuming, and more costly. Machine Learning for Music & Media (MLMM), a company founded by a schooled musician, recording artist, and Machine Learning-engineer, dedicates itself discovering the hidden gems in a convoluted market. One tool that will help in this endeavor is the Song Popularity Predictor (SPP) by predicting a (new) song’s popularity based on its extracted musical attributes, such as danceability, energy, tempo, and much more.

## Application Benefits

The SPP will benefit both the music industry and individual creatives:

* Unparalleled efficiency – sift through 80,000 songs in less than an hour and discover what new songs have the potential to be popular.
* Reduced costs – Piles of demo tracks on desks belong to the past, there’s no need wasting money on unfruitful scouting efforts, and the application will take over a lot of work previously done by humans, implicating a reduction in the need for manpower.
* Great potential for creative human & AI collaboration - Human perception is highly subjective and may be very different from what the AI thinks. As this field develops, humans can collaborate with AI to find new features that make a song popular and to adapt to current musical trends. On a personal level, this tool will enable songwriters to check the quality of their new projects in the light of what the model thinks the current trends are and use SPP’s advice to enhance their product.

## Application Description

The application will be a web application dashboard that is freely accessible to anyone in your company worldwide. Its features will be:

* The ability so set 13 different audio features such as duration, tempo, key, and loudness.
* A visual representation of the data input by the user.
* A song popularity prediction, having three possible outcomes: unpopular, popular, and very popular.

In addition, the Jupyter Notebook holding the data that the Machine Learning (ML) model will be trained on will be included. Just as musical trends evolve, the model will have to evolve during future iterations of the project. At the model’s core sits a Random Forest algorithm that will predict a song’s popularity with at least 80% score on precision, recall, and f1-score. These scores will only increase the more the song database grows. As the database grows, it is important that the data can be understood in a visual way. This can all be done using the Jupyter Notebook, where Python libraries such as Sci-kit Learn, Pandas, Numpy, Plotly, Seaborn, and Matplotlib help with data loading, cleaning, visualizing, feature engineering, preprocessing, and making predictions.

The web application will be built with Streamlit, which turns Python scripts into web applications in minutes.

## Data Description

The dataset used for training and testing the Random Forest model (Spotify and Genius Track Dataset) is publicly available on Kaggle.com. It is updated and maintained periodically. The csv file, loadable using Pandas, can be found and downloaded at the following URL: <https://www.kaggle.com/datasets/saurabhshahane/spotgen-music-dataset/download?datasetVersionNumber=328>.

## Objective and hypotheses

The SPP’s objectives are to save time, save costs, and save frustration by taking over tedious tasks. The User Interface (UI) will be intuitive and easy to understand. The UI can be used to input song features and will output a prediction for its popularity with an accuracy and recall of at least 80%: unpopular, popular, or very popular.

If the company uses the Song Popularity Predictor consistently, then in a year from now you will see a reduction of at least 50% in time passed between release of a song and you discovering it, an increase of 20% in revenue, and a minimum 25% increase in productivity.

## methodology

The CRISP-DM Agile methodology and principles will be applied to the implementation of this project. The codebase will be developed in sprints. That way the application can be improved incrementally as data understanding grows and backtracking is possible when necessary.

* 1. *Business Understanding:* In this phase, we must work towards an understanding among leadership, executive staff, and other managing staff about why this project is needed. What worked in the last century, doesn’t necessarily always work in this one. To adapt to the times and ensure growth for the future, the company needs the help of ML-tools.
  2. *Data Understanding:* In this phase, we will collect, analyze, and visualize the data set using a Jupyter Notebook. Visualizing the data will prove useful in understanding the data and identify relationships (if any). For example, is there a relationship between popularity and the level of positivity in a song? Can we identify a relationship between the tempo of a song and its popularity?
  3. *Data Preparation:* In this phase, we prepare the final data set for modeling. This will be done by selecting the data we want and cleaning it from data that we don’t want. We are solely zooming in on 13 song attributes, namely: acousticness, danceability, duration (in ms), energy, instrumentalness, key, mode, liveness, loudness, speechiness, tempo, time signature, and valence. Even though one can argue an artist name can boost popularity of a song, external factors like that are outside the scope of this project.
  4. *Modeling:* In this phase, we will build and assess a classification model (like a Random Forest Classifier) that will help us achieve the application’s goals. We will split the dataset 80/20; 80% of the dataset will be used for training the model and 20% will be used for testing.
  5. *Evaluation:* In this phase, we will determine how the model meets the goals and objectives. Is the model making correct predictions? Is it ready for deployment or do we need to start another iteration to improve it? Was anything overlooked?
  6. *Deployment:* If all previous phases have been completed to satisfaction, we will be ready for deployment. The application will be deployed using Streamlit, making it available in a web browser to all authorized employees anywhere in the world. Feedback will be collected to improve the current working version and ideas will be collected to identify future projects that build on this one.

## Funding Requirements

Please refer the table for initial project funding estimates:

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Cost/hour | Total hours | Total cost |
| Planning and Design | $100 | 30 | $3,000 |
| Development | $200 | 40 | $8,000 |
| Documentation | $150 | 30 | $4,500 |
| Design Review and Determining Future Work | $150 | 30 | $4,500 |
| Overhead Costs (20%) |  |  | $4,000 |
| TOTAL: | | | $24,000 |

## Stakeholder impact

There are two main groups of stakeholders:

* The music industry
* The creatives (songwriters and composers)

The impact on both groups is a little different, but sometimes related. For the music industry:

* Improved efficiency. More songs can be analyzed in much shorter time.
* Increased profitability by lowering operational costs.
* Faster time to discovery. Discover talent and potential popular songs quicker than the competition.
* Increased popularity success rate. Men lie, women lie, but numbers don’t. The ratio of released songs by the label versus the number of actual popular songs will increase.

For creatives:

* Easy and intuitive to use – the SPP is a hands-on tool for suggestions on how to improve a new song.
* Confidence boosting – data backed confirmation to boost confidence levels.
* Education – Explore what makes other songs popular and what doesn’t.

## Data Precautions

The data itself is publicly available either on Kaggle.com or via the Spotify API. This data is not considered sensitive or protected.

## Developer’s expertise

Two additional developers will be added to the existing development team that consists of a Python programmer and a machine learning engineer. The current team members combined have over 12 years of experience.  
The new members will be required to be strong Python programmers, have experience in web development and Streamlit, be flexible, and comfortable learning new technologies. Experience in deploying applications to services like AWS and Heroku are preferred.

# B. Technical Proposal

## Problem Statement

With over 80,000 tracks being uploaded to major digital service providers each day the music industry market has become convoluted over the past years. Finding new talent and catchy songs has become more challenging, more time-consuming, and more costly. The challenge is to utilize Machine Learning as a tool to take on this challenge and to do this work faster, cheaper, and more efficiently than humans ever could. One tool that will help discover the gems in a convoluted market is the Song Popularity Predictor (SPP), a machine learning model that predicts a (new) song’s popularity based on its extracted musical attributes, such as danceability, energy, tempo, and much more, through a user-friendly UI in a web browser environment to enable a high level of accessibility.

## Customer Summary

This application will satisfy the business needs of those in the company charged with promoting existing artists, those working to find new, hidden talent and the company’s songwriters/composers.

* The app will be a tool to help pick the next song release for an existing artist, simply by comparing the release options’ attributes and picking the one with the highest popularity prediction.
* Those with the responsibilities of finding and signing new talent can input the song’s attributes into the tool to help them decide who is going to be signed on a record deal next.
* The company’s songwriters and composers will use the app to confirm if their latest written songs are going to be popular or not. They’ll also be able to identify what areas in the song can be improved to increase its popularity potential.

No special skills are needed to use the application, thanks to the user-friendly UI for which only basic web navigation skills are required.

## Existing System analysis

Currently, the company has no machine learning tool in place. It only has access to Spotify for Artists, which gives basic information regarding listener’s demographics and track performance. What songs to release next is largely based on a hunch and lots of manpower is needed to filter through the more than 80,000 new tracks being uploaded to major distribution platforms each day. Songwriters and composers are only as good as their last song. Upon completion of this project, the next song to release is an educated guess to say the least, employees can filter through new song releases faster than before, and songwriters/composers will have the ability to score their compositions and make adjustments if necessary.

## DATA

The dataset used for training and testing the initial model is in .csv format, clean, and publicly available on Kaggle.com. Future data will be collected through the Spotify API, converted to .csv format, and run through the SPP app. At any time can new data be added to the original dataset to retrain and update the prediction model. New data will most likely always be clean, as the model uses the same song attributes that Spotify outputs through their API. Newly acquired data will have to be cleaned from unnecessary attributes before adding to the existing dataset. These attributes are analysis\_url, id, track\_href, type, and uri.

## Project Methodology

To manage this project, the Agile methodology will be used.

1. Concept: here, we will determine and document the project’s scope. After gathering key requirements, documentation will be produced to outline them, including what features will be developed and what the end results should be. Estimated time requirements and costs will be documented as well.
2. Inception: here the design process is started. A UI mockup will be created and project architecture will be built.
3. Iteration: this is the construction phase, where the bulk of the work will be carried out. The design will be turned into code, while taking into consideration all the requirements. At the end of the first sprint, a prototype product will be produced with minimal functionality. Additional features and adjustments will be added in later iterations. Being able to show incremental improvements will help with client satisfaction.
4. Release: After performing full quality assurance through testing, the product will be ready for deployment. Potential bugs or defects will be addressed swiftly. More documentation will be produced to help with the training of users.
5. Maintenance: The web application will now be fully available to users and the Jupyter Notebook will be ready to be used by those responsible for project maintenance. Our development team will provide ongoing support to resolve any new bugs. When necessary, additional training can be done to ensure all users know how to use the application. Over time, new iterations can be performed to enhance the existing product with additional features, upgrades, and use-cases.

## Project Outcomes

The project outcomes consist of two distinct categories: Project Deliverables and Product Deliverables.

**Project Deliverables**:

* Milestones schedule: to keep track of project timeliness and budget an in-depth schedule of the project milestones will be required.
* Test plan: a test plan outlining pre-train tests to catch bugs before running the model, post-train tests to check whether the model performs correctly, and directional tests to check on how changes in input affect the model prediction.
* Pseudo-code: pseudo-code shall be created to outline the data analysis, data preparation, data processing, training, and evaluation phases. The same shall be done for the web application, including a UI mockup.
* Coding Requirements: a list of all environment requirements and Python libraries needed to run the Jupyter Notebook and the web application.

**Product Deliverables**:

* + Jupyter Notebook – a Jupyter Notebook shall be created that analyzes, visualizes, and process the dataset. Finally, it will train explore the performance of several different models, after which the best model will be chosen to be deployed for use in the web application.
  + Maintenance tools – the Jupyter Notebook has all the necessary tools to reevaluate and adjust the code to generate better model performance, try new algorithms, and add more attributes to be taken into consideration by the model.
  + Web Application – a web application with a user-friendly UI deployed with the help of Streamlit. The user will be able to input 13 different song attributes to generate a popularity prediction.
  + Restricted Access optionality – when desired, access to the app can be restricted by only allowing access to people with certain authorized email addresses. The default app setting currently is to grant access to anyone that has the link to the web app. This setting can be adjusted at any time.

## Implementation plan

The CRISP-DM Agile methodology and principles will be applied to the implementation of this project. The codebase will be developed in sprints. That way the application can be improved incrementally as data understanding grows and backtracking is possible when necessary.

* 1. *Business Understanding:* In this phase, we must work towards an understanding among leadership, executive staff, and other managing staff about why this project is needed. Techniques and workflows that worked in the last century, don’t necessarily work in this one as well. To adapt to the times and ensure growth and sustainability for the future, the company needs the help of ML-tools.
  2. *Data Understanding:* In this phase, we will collect, analyze, and visualize the data set using a Jupyter Notebook. Visualizing the data will prove useful in understanding the data and identify relationships (if any). For example, is there a relationship between popularity and the level of positivity in a song? Can we identify a relationship between the tempo of a song and its popularity? To accomplish this, we will use barplots, boxplots, scatterplots, etc. using the seaborn, plotly, and matplotlib libraries.
  3. *Data Preparation:* In this phase, we prepare the final data set for modeling. This will be done by selecting the data we want and cleaning it from data that we don’t want. We are solely zooming in on 13 song attributes, namely: acousticness, danceability, duration (in ms), energy, instrumentalness, key, mode, liveness, loudness, speechiness, tempo, time signature, and valence. Even though one can argue an artist name can boost popularity of a song, external factors like that are outside the scope of this project.
  4. *Modeling:* In this phase, we will build and assess several classification models (like Logistic Regression, Random Forest Classifier, and boosting models like Adaboost) that will help us achieve the application’s goals. All models are examples of supervised learning. We will split the dataset 80/20; 80% of the dataset will be used for training the model and 20% will be used for testing.
  5. *Evaluation:* In this phase, we will compare models and determine which one performed the most desirable by looking closely at the confusion matrices and accuracy reports. Then we will decide if the chosen model is ready for deployment. Perhaps another iteration is needed to improve it. These improvements can be achieved by doing better data resampling, more feature engineering, trying different encoding techniques, or doing more transformations such as log transformation.
  6. *Deployment:* If all previous phases have been completed to satisfaction, we will be ready for deployment. The application will be deployed using Streamlit, making it available in a web browser to all authorized employees anywhere in the world. Feedback will be collected to improve the current working version and ideas will be collected to identify future projects for improvement that build on this one.

## Evaluation plan

Model accuracy validation and verification is done through ML accuracy reports and by using confusion matrices. Throughout the development process, the Jupyter Notebook and web application will be subject to continuous testing and evaluation, enforced by the Agile development process, which allows us to resolve issues and course correct quickly when issues are discovered.

## Resources and Costs

### Programming Environment

The tools and environment needed for this project are:

* Any computer or device able to run a web browser in order to use the web application.
* Jupyter Notebook (including software like Anaconda)
* Python (version 3.9 at minimum), including all required libraries (Numpy, Pandas, Seaborn, Matplotlib, ipywidgets, pickle, imblearn, and sci-kit learn)
* Git version control

### Environment costs

|  |  |
| --- | --- |
| Resource | Cost |
| 2 additional PC’s for newly hired developers | $1,000 |
| TOTAL: | $1,000 |

### Human Resource Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Cost/hour | Total hours | Total cost |
| Planning and Design | $100 | 30 | $3,000 |
| Development | $200 | 40 | $8,000 |
| Documentation | $150 | 30 | $4,500 |
| Design Review and Determining Future Work | $150 | 30 | $4,500 |
| Overhead Costs (20%) |  |  | $4,000 |
| TOTAL: | | | $24,000 |

## Timeline and Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Start | End | Tasks |
| 1 | Date  10/18/2022  10/24/2022  10/27/2022 | Date  10/21/2022 10/26/2022  10/28/2022 | **Planning and project setup:**   * Define task and scope requirements. * Produce documentation concerning application features and requirements. * Document time requirements and estimates. |
| 2 | Date  10/31/2022 11/2/2022 11/7/2022 11/10/2022 | Date  11/1/2022 11/4/2022 11/9/2022 11/11/2022 | **Data collection, preparation, and visualization:**   * Document data surface properties. * Identify data relationships. * Prepare data set for modeling. * Revisit Sprint 1. |
| 3 | Date  11/14/2022 11/16/2022 11/18/2022 11/23/2022 12/05/2022 | Date  11/15/2022 11/17/2022 11/22/2022 12/02/2022 12/09/2022 | **Model training, testing, and validation:**   * Build the different ML models. * Train all the ML models. * Test all the ML models. * Revisit Sprint 1 & 2.   Complete and finalize all deliverables. |
| 4 | Date  12/12/2022  12/15/2022  12/20/2022 | Date  12/14/2022  12/19/2022  12/23/2022 | **Evaluation:**   * Train user on how to use the Jupyter Notebook and web application. * Use and test the system to see if it meets business requirements.   Determine the next steps. Revisit Sprint 1, 2, & 3 if necessary. |
| 5 | Date  12/27/2022 01/02/2023 01/09/2023 | Date  12/30/2022 01/06/2023 ongoing | **Deployment:**   * Produce final report and presentation. * Produce a full project review.   Monitoring and Maintenance. |

# C. Application Files

Please refer to the following Jupyter Notebook file and web app Python script for the project code:

* C964 - Song Popularity Predictor.ipynb (also viewable at <https://github.com/jonivanrossum/C964/blob/main/C964%20-%20Song%20Popularity%20Predictor.ipynb>)

OR

* C964 - Song Popularity Predictor.pdf (also viewable at <https://github.com/jonivanrossum/C964/blob/main/C964%20-%20Song%20Popularity%20Predictor%20-%20JupyterLab.pdf>)
* webapp.py (also viewable at <https://github.com/jonivanrossum/C964/blob/main/webapp.py>)

Try out the web app at:

* [https://jonivanrossum-c964-webapp/-0yn76e.streamlitapp.com/](https://jonivanrossum-c964-webapp-0yn76e.streamlitapp.com/)

# D. Post-Implementation Report

## Project Purpose

## Datasets

## Data Product Code

## Hypothesis Verification

## Effective Visualizations and Reporting

## Accuracy Analysis

## Application Testing

## Application Files

## User’s Guide

## Summation of Learning Experience

# E. sources

Leight, E. (October 11, 2022*). Too Many Songs, Not Enough Hits: Pop Music Is Struggling to Create New Stars.* Billboard.com. Retrieved from: <https://www.billboard.com/pro/new-music-tiktok-artist-development-suffering/>