Assignment 2: Data Analytics

188.429 Business Intelligence (VU 4,0) – WS 2024

Tuvshin Selenge*
TU Wien
Vienna, Austria
e11815791@student.tuwien.ac.at

Simon Tritscher
TU Wien
Vienna, Austria
e11701226@student.tuwien.ac.at

Figure 1: This assignment uses a dataset of 30000 Spotify songs.

1 Introduction

In this assignment, we aim to address real-world data analytics challenges through the application of machine learning techniques. By training models on our dataset, we strive to develop robust solutions that effectively tackle these challenges. Through careful model creation, testing, and evaluation, our goal is to generate meaningful insights and results that can be leveraged by potential stakeholders to inform decision-making and drive value.

2 Business Understanding

2.1 Data Source and Scenario

We are working with a Kaggle dataset compiled using the official Spotify API. This dataset includes over 28,000 unique tracks from the 2010s, covering a broad spectrum of music from around the world.

Analyzing this dataset aims to gain insights into the music industry, uncover trends in listener preferences, and identify shifts in musical tastes over the decade. By understanding which genres, song lengths, and attributes contribute to a song's success, we can predict the characteristics that might make future tracks popular. This information could be valuable for musicians and producers, helping them create music that resonates with audiences and avoids becoming "one-hit wonders."

2.2 Business Objectives

For stakeholders in the music industry, such as major record labels, gaining new listeners and staying informed about the latest trends is crucial. Sony Music Entertainment Germany GmbH, one of the leading players in the industry, experienced a significant decline in net asset value from 413 million in 2022 to 356 million in 2023 [2]. This drop could have serious implications, including budget reductions and decreased investment capacity, ultimately limiting innovation and risking stagnation.

Given these challenges, our analysis and predictions aim to provide Sony Music with strategic insights into industry trends and help shape an effective path forward. By identifying opportunities for growth and staying ahead of market shifts, we can support Sony Music in adapting to changing dynamics and improving its overall strategy.

2.3 Business Success Criteria

We will measure our success after two years, as this is a reasonable timeframe to evaluate whether our predictions and the actions based on our recommendations have had an impact on revenue generation. Specifically, we will track whether our insights regarding trending genres and music length have influenced the creation of new songs and releases. Success will be determined by assessing if these new releases, based on our recommendations, contribute to an increase in annual revenue and listener engagement. Each success criterion will be tied to the overarching business objectives of increasing market share, fostering innovation, and sustaining financial growth.

2.4 Data Mining Goals

According to our business objectives and success criteria, we want to use the given data to predict the characteristics of future popular songs. This results in mainly the following data mining goals.

Trend Analysis. As the given dataset contains songs from the 2010s, we can work on discovering different trends throughout the decades. This will give us a better oversight in what song characteristics were popular and in the beginning of the given decade, while then following trendlines across the whole ten years. Ultimately,

*Both authors contributed equally to this assignment. Tuvshin Selenge has the role of person A, while Simon Tritscher has the role of person B.

this should provide insights in how listener behavior changes over time.

Popularity Prediction. Given the characteristics of a new song, we aim provide a regression model which predicts the popularity of that song with a high accuracy. This will help stakeholders in gaining a better overview in how their song will perform once it will be released to consumers. Additionally, we aim to get insights from this model on which parameters are key in influencing song popularity. In providing this information to our musicians and producers, they will be able to "tune" their new productions specifically to increase their potential popularity.

2.5 Data Mining Success Criteria

Having our goals defined, we can now define related success criteria according to which we can track our progress of this project. Our first success criterion is to extract reasonable insights from our given dataset. This includes being able to show up valuable and insightful information for our stakeholders, including trends in listener preferences and the most important characteristics driving song popularity. As for our prediction model performance, we define success in achieving a RMSE below 10 for our predictions on a test set.

2.6 AI Risk Aspects

While we are dealing with non critical nor personal data and our project scope is limited, there should not be any major AI risks involved. However, there could be still several issues with the dataset and the predictions thereof, which have to be actively monitored and worked on. For one, the dataset does not show which region this popularity score is about. This is a big issue, because listeners around the world have different preferences, which is not reflected here. Also, a high popularity of a song should not always reflect the aims of every artist, there is a thriving music community which produces indie music not catered to the general consumers. Furthermore, the model may fail to predict new trends as it is based entirely on historical data. Lastly, it has to be transparent in how the model comes to conclusions in order to be able to reduce bias and to explain what the important factors in songs are.

3 Data Understanding: Data Description Report

3.1 Attributes Description

The dataset predominantly consists of nominal and interval attributes, with a few ratio attributes. Most attributes are nominal, such as track_id, name, artist, and popularity, which are listed in Table 1. Interval attributes include danceability, energy, and valence, all measured on a scale from 0 to 1. Danceability indicates how suitable a track is for dancing, based on factors like tempo and beat, while energy reflects the intensity of a track, taking into account elements such as loudness, noise, and speed. Valence measures the emotional positivity of a track, ranging from 0 (sad) to 1 (happy).

The dataset also includes two ratio attributes: tempo and duration. Tempo provides the estimated beats per minute (BPM) for each track, and duration represents the length of the track in milliseconds.

3.2 Statistical properties and Correlation analysis

The dataset contains over 30,000 songs. From the statistical properties presented in Table 2, we can observe that most of the songs are both danceable and energetic. The average danceability score is approximately 0.65 (on a scale from 0 to 1), and the mean energy level is 0.70, indicating a tendency toward high-energy music. The dataset also includes some unusually short songs, with durations as low as 4,000 milliseconds (equivalent to 4 seconds), which could be potential outliers.

In Figure 2, our correlation analysis reveals that there is a strong positive correlation between energy and loudness, while energy shows a negative correlation with acousticness. Apart from these relationships, there are no other significant correlations in the dataset worth noting.

3.3 Data Quality and Visual exploration

In our analysis of data quality and visual exploration, we examined the quantitative account of missing values. Missing values were present for only three attributes: track_name, track_artist, and track_album_name, as shown in Table 3. The other attributes had reasonable values, but further inspection revealed that these 15 missing entries were linked to Latin or Latin American roots. It is possible that these records contained special characters or accents that could not be correctly interpreted by the CSV file. Given the minimal impact, these five rows will be cleaned accordingly.

In our visual exploration, we created several plots showing the distribution of energy, track popularity, duration, and different genres. The energy distribution was left-skewed, indicating a higher tendency towards energetic songs. The track popularity plot showed a high frequency of less popular songs with low scores, while the rest of the scores were fairly evenly distributed. The duration attribute exhibited a clear normal distribution, with a maximum value of 50,000 ms and a minimum of 4,000 ms, as reflected in the statistical properties. Figure 6 displays the various genres, which are relatively evenly represented. EDM had the highest percentage at 18.4%, while rock had the lowest at 15.1%.

3.4 Risks and Biases

As we understand the data, there is no personal information included that poses a significant risk—only song titles and related metadata from Spotify are involved. However, biases could naturally be present in the dataset. For example, some songs may achieve higher scores due to the influence of social media or time-based biases associated with Spotify usage. Our dataset spans from early 2010 to the end of the decade, a period during which social dynamics have shifted considerably, potentially impacting song popularity. This is precisely what we aim to analyze: understanding how trends evolve over time and predicting future trends based on historical data.

4 Data Preparation Report

4.1 Pre-processing of Dataset

Most columns are already in a suitable format for our regression model, however there are some columns which we will not need.

Table 1: Track and Playlist Attributes

Variable	Attribute Type	Description
track_id	Nominal	Song unique ID
track_name	Nominal	Song name
track_artist	Nominal	Song artist
track_popularity	Interval	Song popularity (0-100)
track_album_id	Nominal	Album unique ID
track_album_name	Nominal	Album name
track_album_release_date	Nominal	Album release date
playlist_name	Nominal	Playlist name
playlist_id	Nominal	Playlist ID
playlist_genre	Nominal	Playlist genre
playlist_subgenre	Nominal	Playlist subgenre
danceability	Interval	Suitability for dancing (0.0 - 1.0). Danceability describes how suitable a track is for dancing based on tempo, rhythm stability, beat strength, and overall regularity.
energy	Interval	Intensity and activity level (0.0 - 1.0). Energy is a measure from 0.0 to 1.0 representing intensity and activity. Energetic tracks are fast, loud, and noisy, while calmer tracks score lower.
key	Nominal	Overall key of the track (0 = C, 1 = C#/Db, etc.) The estimated overall key of the track using Pitch Class notation.
loudness	Interval	Average track loudness in dB (-60 to 0)
mode	Nominal	Mode indicates the modality of the track: 1 for major and 0 for minor.
speechiness	Interval	Speechiness detects the presence of spoken words in a track. Values range from 0.0 to 1.0. Tracks with values above 0.66 are mostly spoken.
acousticness	Interval	Confidence the track is acoustic (0.0 - 1.0)
instrumentalness	Interval	Likelihood track is instrumental (0.0 - 1.0)
liveness	Interval	Presence of audience (0.0 - 1.0)
valence	Interval	Valence measures the musical positiveness conveyed by a track, ranging from 0.0 (sad, depressed) to 1.0 (happy, cheerful).
tempo	Ratio	Estimated tempo in BPM
duration_ms	Ratio	Duration in milliseconds

Table 2: Statistical Properties of the Dataset

Attribute	Count	Mean	Std	Min	25%	50%	75%	Max
track_popularity	32833	42.48	24.98	0	24	45	62	100
danceability	32833	0.65	0.15	0.00	0.56	0.67	0.76	0.98
energy	32833	0.70	0.18	0.00	0.58	0.72	0.84	1.00
key	32833	5.37	3.61	0	2	6	9	11
loudness	32833	-6.72	2.99	-46.45	-8.17	-6.17	-4.65	1.28
mode	32833	0.57	0.50	0	0	1	1	1
speechiness	32833	0.11	0.10	0.00	0.04	0.06	0.13	0.92
acousticness	32833	0.18	0.22	0.00	0.02	0.08	0.26	0.99
instrumentalness	32833	0.08	0.22	0.00	0.00	0.00	0.00	0.99
liveness	32833	0.19	0.15	0.00	0.09	0.13	0.25	0.99
valence	32833	0.51	0.23	0.00	0.33	0.51	0.69	0.99
tempo	32833	120.88	26.90	0.00	99.96	121.98	133.92	239.44
duration_ms	32833	225799.81	59834.01	4000	187819	216000	253585	517810

There are three columns containing IDs (track_id, track_album_id, playlist_id), these make no sense to use in our model as these are mostly unique and add no real value to our usecase. Also, most columns containing strings such as artist, song or album name also do not contain useful information (namely track_name, track_artist, track_album_name, playlist_name). Only two columns about the genre of the song could be useful (playlist_genre, playlist_subgenre),

Table 3: Missing Values in Dataset

Feature	Missing Values
track_id	0
track_name	5
track_artist	5
track_popularity	0
track_album_id	0
track_album_name	5
track_album_release_date	0
playlist_name	0
playlist_id	0
playlist_genre	0
playlist_subgenre	0
danceability	0
energy	0
key	0
loudness	0
mode	0
speechiness	0
acousticness	0
instrumentalness	0
liveness	0
valence	0
tempo	0
duration_ms	0

so we drop all mentioned columns from the dataset except for these two.

There are six different playlist genres in the dataset. Additionally, there exist 24 different playlist subgenres. These may be of high informational value for our model, but they are initially in string format. For our preprocessing, we use one-hot encoding to create additional attributes (boolean) within the dataset containing the information of genre and subgenre.

Also, there is a release date in the form of 'yyyy-mm-dd'. We convert that into first a pandas datetime format. The exact date may be too specific information for our usecase, so we only extract the release year from that column and drop the month and day data.

In the data understanding phase, we found out that a small amount of the data contained missing values. These missing values however only were contained within the columns we dropped during the preprocessing phase. Therefore, the problem with missing values solved itself and we did not even need to drop the lines containing the missing values from the dataset.

After all above steps, the dataset should now be sufficiently preprocessed for the initial model creation phase.

4.2 Potential for Derived Attributes

There is potential for derived attributes and we use them during our data preprocessing phase.

Datetime Features. One column contains the release date of the track in ISO8601 format. This exact date may be too specific to use, as we want to be able to generalize within our regression model. However, we can just extract the year from the data and use this instead of the full date. This should ultimately help in improving the predictive performance of our model.

Categorical Encoding. While we should dismiss the song titles and artist and playlist names as these are arbitrary and contain little informational value, we should use the genres which are limited in their amount and should be useful as categorical data. There are six genres and 24 subgenres. We aim to get the best performance for our model by using one-hot encoding, which creates one column for each category and uses true/false for marking what category is applicable for each row.

We are currently not sure if also using the subgenres column improves or worsens model performance. As this column consists of 24 different categories, some categories may appear much more often than others. This could be an issue in the models' generalization ability. Our current plan is to train the model both with and without this column and then decide according to the results if we continue with that column or leave it out.

4.3 Potential for External Data Sources

During our initial assessment phase we already mentioned the missing information about regionality within the dataset. We are not sure if the popularity score of the songs within the dataset are a global score or if they are taken from a specific region. Since we want to support an European company with this work, it would be crucial to have the correct data for this market. If the dataset has for example a popularity score taken from Latin American consumers, the informational value would be low and the model would not provide information tailored to European listeners. Even worse,

providing such results may even be damaging to our stakeholders as they may produce music that European listeners do not want and would not listen to. Therefore, getting additional regional data for our dataset would be useful or even a must in a real world scenario. For the sake of our project and due to the fact we could not find such information for the data we used, we will just assume that this data is tailored to the European market.

Other than that, the data should be very sufficient for our use case and we can currently not think of any further data sources needed to fulfill our business objectives and data mining goals.

5 Modeling and Evaluation

5.1 Modeling

The dataset is split in two stages: first into a train-validation set and a test set, and then further into separate training and validation subsets. Before splitting, the target variable is removed to isolate the features for preprocessing and avoid data leakage.

During preprocessing, several steps are undertaken to prepare the data for modeling:

(1) **Handling Missing Values:** Rows with missing data are removed to ensure data consistency and reliability.
(2) **Date Conversion and Feature Extraction:** The release date of tracks is converted to a datetime format, and the release year is extracted as a new feature. This transformation enables the model to leverage temporal information effectively.
(3) **Dropping Irrelevant Columns:** Non-essential columns, such as IDs, track names, and playlist names, are removed to reduce noise and focus on relevant features.

These preprocessing steps are performed separately on the training, validation, and test sets to maintain consistency and prevent any data leakage. By isolating each set, we ensure that the evaluation metrics provide an unbiased measure of model performance.

In the initial stages of our modeling process, we experimented with three models: Multilayer Perceptron (MLP), Support Vector Machines (SVM), and Random Forest. However, we soon encountered computational limitations. The processing requirements of these models exceeded our available computing power, leading to prolonged runtimes without yielding results. Consequently, we revised our approach and chose for models that are computationally less intensive yet effective for large datasets. Our final selection includes:

- **Ridge Regression**
- **Stochastic Gradient Descent (SGD)**
- **Random Forest Regressor**

These algorithms were chosen for their balance between computational efficiency, scalability to large datasets, and ease of implementation. Below, we provide a brief overview of each method:

5.1.1 Ridge Regression. Ridge regression is a linear model that enhances ordinary least squares by incorporating a regularization term. This term minimizes the residual sum of squares between the observed and predicted target values while penalizing the magnitude of the model's coefficients. The penalty, proportional to the sum of squared coefficients (L2 regularization), helps reduce overfitting by shrinking coefficients toward zero, thereby improving model generalization.

5.1.2 Stochastic Gradient Descent (SGD). Stochastic Gradient Descent is an iterative optimization algorithm that updates model parameters based on the gradient of the loss function calculated for individual samples. It uses a decreasing learning rate schedule to ensure convergence over time. Regularization can be applied during the optimization process to shrink parameters toward zero, using the squared Euclidean norm (L2), the absolute norm (L1), or a combination of both (elastic net).

5.1.3 Random Forest Regressor. Random Forest is an ensemble learning method that constructs multiple decision tree regressors on random subsets of the dataset. It combines the predictions of these trees through averaging, thereby enhancing predictive accuracy and reducing the risk of overfitting. The approach leverages the diversity of individual trees to achieve robust performance.

By leveraging these algorithms, we ensured that our data mining tasks were computationally feasible and well-suited for the size and complexity of our dataset, without compromising the quality of the results.

5.1.4 Hyper-parameter Tuning. For our hyper-parameter tuning, we opted for a computationally efficient approach by using RandomizedSearchCV, which provides faster results compared to GridSearchCV. Initially, we had implemented GridSearchCV, but this method proved to be too resource-intensive and failed to produce results within a reasonable runtime.

RandomizedSearchCV optimizes the parameters of an estimator by performing a cross-validated search over specified parameter distributions. Unlike GridSearchCV, which exhaustively evaluates all possible parameter combinations, RandomizedSearchCV samples a fixed number of parameter settings from the specified distributions. The number of settings to evaluate is controlled by the `n_iter` parameter, making the process more efficient for large datasets and complex models.

The best hyper-parameters identified through this process, along with their corresponding values, were recorded for subsequent use in the later stages of the analysis.

5.2 Results from the Validation Set

The performance of our models on the validation set is summarized in Table 5. The metrics include the Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2 Score). These results provide a comparison of the models predictive capabilities.

Based on our results, we concluded that the Random Forest Regressor provided the best performance among the evaluated models. Therefore, we selected it for further use in subsequent analyses. The optimal hyper-parameters identified through our random search for the Random Forest Regressor are as follows:

- **bootstrap:** True
- **max_depth:** 30
- **max_features:** None
- **max_samples:** 0.5

- **min_samples_leaf:** 2
- **min_samples_split:** 2
- **n_estimators:** 200

These parameters will be used in later stage of our analysis to compare the Random Forest Regressor performance against the baseline models.

Table 4: Model Performance Comparison on Validation Set

Model	MSE	MAE	R^2 Score
Random Forest Regressor	425.88	16.98	0.316
Ridge Regression	522.37	18.87	0.161
SGD Regressor	624.27	20.65	-0.002

5.3 Final Model Results

5.3.1 Predictive Qualities of our Model. After deciding on the optimal model and fitting hyperparameters for that model, we could take further steps and check the performance of our fitted model against the test set. Afterwards, we fitted the model again on the full training plus validation set (to use all of our available training data) and predicted against the test set. The results are as follows.

Table 5: Final Model Performance Comparison

Model	MSE	MAE	R^2 Score
Fit on Validation Set	431.92	16.72	0.306
Fit on Full Train Set	428.54	16.58	0.312

The results show that using more data to fit the model does improve the prediction performance, however just very slightly. The results of the latter fit are final results of our model. These show that while the model should have some predictive qualities, the error rate is quite high. A mean absolute error of 16.58 shows that the average prediction is off by 16.58 on a popularity scale of 0-100. In a later stage, we check if these results perform better than a trivial and a random predictor baseline and also compare them to similar works.

5.3.2 Further Model Insights. Next to the insights gained during the data exploration phase we could also further analyze our trained model to get additional insights about the data. Using random forest as model, we can easily extract the feature importances most relevant in predicting the popularity score.

We can see, that album release year has the most influence in predicting the popularity score. Initially this was not that clear to us, so we plotted the popularity values against the release year and found out that the songs reached higher popularity scores the more recent they were. Only songs which came out in the last 5-10 years had the opportunity to get a popularity score of above 85. Therefore, the model could probably improve its predictive qualities by looking at the date the song was released.

Other influential attributes were among others loudness, instrumentalness and song duration. In the appendix you can find these most influential attributes plotted against the popularity target variable. While there are generally low correlations to be found, some insights might still be gained here. For example the most popular songs all have a song length within a narrow range.

Table 6: Top 15 Features for Random Forest Regressor

Feature	Importance
album_release_year	0.110893
loudness	0.080742
instrumentalness	0.077507
duration_ms	0.073040
energy	0.072164
tempo	0.071256
acousticness	0.071038
danceability	0.067520
speechiness	0.067156
valence	0.066718
liveness	0.063682
key	0.033631
playlist_subgenre_permanent wave	0.015617
playlist_genre_edm	0.014573
playlist_subgenre_progressive electro house	0.012158

5.4 State of the Art Algorithms using Spotify API Data

We used a dataset generated via the Spotify API and uploaded to Kaggle. After some research we could find many very similar projects to ours, popularity prediction using a collection of Spotify songs and regression models seems to be a popular student and data science project. However, we could not find any projects using the exact same excerpt of songs we used. It seems, that others just used the Spotify API to extract their arbitrary amount of data. Therefore, we could neither find any solid or even peer-reviewed sources, nor any other public repositories or blog posts to directly compare our results and models with.

However, for our comparison we singled out two sources which are most comparable to our project. One blog post on the data science website "towardsdatascience.com" and a project from a person who posted their project and results on a public Github repository.

Blog Post: Predicting Popularity on Spotify—When Data Needs Culture More than Culture Needs Data. In this blog post by Philip Peker, he gave an introduction into a data science workflow using Spotify song data with the goal of popularity prediction. He also tried out different models on the data, but a notable difference to our project is that the dataset he used contained around 587,000 songs instead of our 30,000. After an initial data exploration, he used linear regression, a decision tree and random forest to try to get good results. He achieved the best results with the random forest model, however was not ultimately satisfied with the metrics he achieved. After this, he tried to use classification models to try to classify the predicted songs into different ranges of prediction, which also did not lead to much better results. Finally, he provided some possible reasons as of why predictive models using the variables of the dataset can not perform better than the provided results. [3]

Github Repository: Predicting Spotify Song Popularity. This Github repository from Matt Devor is also one of the more sophisticated projects using Spotify data. After an initial data exploration phase, he tried out linear regression and logistic regression in order to predict the popularity variable. He used around 116,000 songs for this task, and also tried out different techniques such as undersampling to handle certain aspects of his dataset. However, he faced difficulties in predicting the target variable within a reasonable accuracy and his resulting performance metrics are not very good. [1]

5.5 Implemented Baseline models

In order to check if the model achieves better performance than random guesses or other trivial models, we implemented one trivial baseline and one random classifier and compared the results.

Trivial Baseline. For a trivial baseline model we first calculated the mean popularity score for the whole train set. Using just this mean value, we predicted the whole test set and then calculated our performance indicators.

Random Predictor. As the second baseline, we implemented a random classifier which assigned to each prediction of the test set one random value in the range between zero and 100 (the range of the target variables of the train set). Again, we added the results of our performance indicators to our results list for further comparison.

5.6 Comparison with Benchmark and Baseline performances

As a comparison metric we use the RMSE (Root Mean Squared Error). The results are as follows.

Model	RMSE
Random Forest Regressor (our model)	20.70
Trivial Baseline	24.96
Random Classifier	38.96
Random Forest Peker	14.80
Linear Regression Matt	26.31

Table 7: Model Performance Comparison

As we can see, our random forest regression model performs better than both our baselines and also the linear regression model from the Github repository. The best results however are from the blog post. Peker did achieve a significantly smaller RMSE than us. However, since the two compared-to "state of the art" models did use other datasets (and significantly more songs) than us, the RMSE comparison with these two model results does not make that much sense. However, by comparing our random forest model with our baselines makes very much sense and shows that our regressor achieves a (partly) significantly better performance than trivial models or random guesses.

5.7 Does Performance Fulfill Success Criteria?

In our data mining success criteria, we defined two main points. The first criterion is about extracting reasonable insights from the dataset. This criterion was intentionally formulated in a more general way, in order to leave room for a broad range of possible insights. We could fulfill this during our data exploration and also modeling phase, since we successfully extracted important and meaningful insights there (as stated in the results section).

Regrettably, we could not fulfill our second success criterion. This criterion was clearly formulated as that we should get a high prediction performance for our model which achieves a RMSE below 10 for predictions on the test set. Our performance is quite far off, with a RMSE of slightly over 20. Further efforts to improve this score did not help in significantly improving the score. With what we learned during the project phase, we assume that the data we used does not have sufficient predictive qualities which would allow to get a low RMSE score.

5.8 Identified protected attributes

Protected attributes in a data mining context refer to features which are sensitive in a ethical or legal way. Since our dataset only contains attributes about songs, we have no protected attributes to watch out for.

6 Deployment

6.1 Performance Comparison for Business Objectives

In our business objectives we tasked ourselves with supporting a major record label and its stakeholders to identify opportunities for growth and provide them with competitive informational advantage. Our results show that we can definitively provide the company with important insights which possibly help in improving its strategy. However, we could not fulfill everything we set ourselves as goals and criteria. While we can deliver our learned insights, our predictive model does not fulfill our self-set data mining goals and lacks predictive quality to be deployed in its current form. One big concern would be that relying on our model to optimize song production for bigger popularity would mislead artists and producers, since the actual outcome of popularity could differ greatly to the predicted outcome of our model.

Since track popularity may depend on many more attributes and factors than there are available in our used dataset, we assume that a predictive model using just the attributes we have available in our dataset will not be able to achieve satisfying results for our use case. This opinion is also reflected in other works trying to achieve the same goal using just the Spotify dataset—both of our comparison projects also come to that conclusion. In our data exploration we also found out that correlations between the given attributes and the popularity score are very low, further deepening our conclusion that deploying a predictive model in this form does not make much sense.

6.2 Impact Assessment and Ethical Aspects

The deployment of our analysis and insights may not fully represent different music genres. As shown in the figure illustrating the distribution of “playlist genres,” we can observe that most common genres are well-represented. However, music trends can shift significantly over time, and predicting the emergence of new or

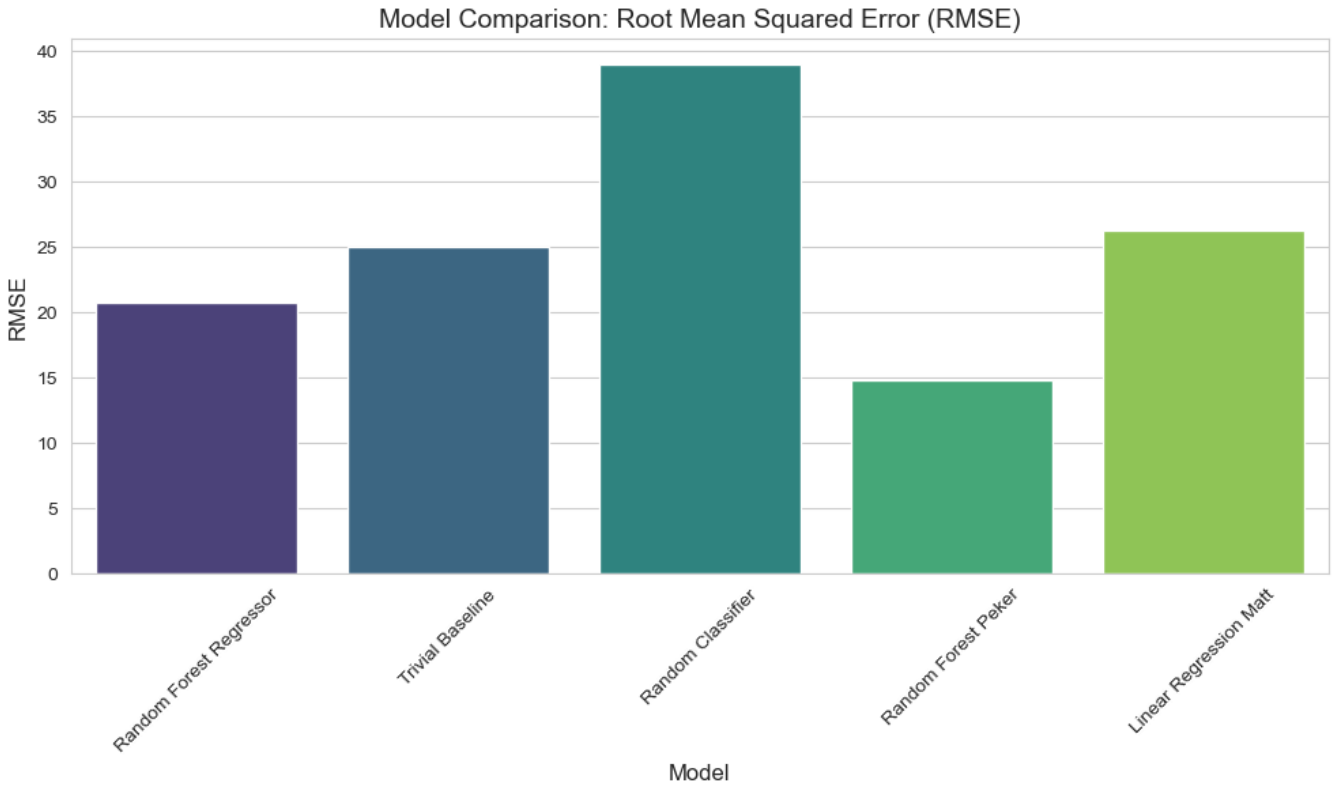

Figure 2: RMSE comparison between model, baselines and SOTA models.

unknown genres is challenging. This limitation arises because our dataset does not include sufficient information about niche genres or provide extensive data for genres that have recently gained popularity, such as the rise of K-pop in recent years.

From an ethical perspective, the dataset's under-representation of lesser-known traditional or national songs is also a concern. These genres appear less frequently in the data, making it difficult to predict their trends accurately. This lack of representation could potentially bias our analysis and limit the applicability of our insights to diverse cultural contexts.

6.3 Reproducibility Aspects

To ensure reproducibility, we have thoroughly documented our process in both the README file and the code associated with this assignment paper, which are available in our publicly accessible GitHub repository [4]. The documentation includes clear instructions on how to replicate the analysis, details about the dataset and its origin, and the specific versions of the packages used. Additionally, we have included a Creative Commons license, "CC0 1.0 Universal (Public Domain Dedication)," to ensure the rightful and unrestricted use of our analysis by others. These measures aim to provide a transparent and replicable framework for others to reproduce our results with accuracy and consistency.

7 Summary and Lessons Learned

During the course of this project we went through the whole CRISP-DM Process with a self chosen dataset. We chose a dataset containing data about songs exported from Spotify via the Spotify Web API. Having the dataset defined, we then created a fictional business case and defined clear goals we wanted to achieve. After performing an exploratory data analysis and then trying out a variety of models for our prediction task, we compared the results with comparable works from others and with trivial and random baselines. Finally, we interpreted our results and compared them to our initial expectations.

7.1 Exercise Feedback

This assignment was quite insightful, as it allowed us to perform an in-depth analysis using Spotify data. The results were somewhat disillusioning, as we initially believed we could apply more performant models, such as SVM or MLP, to our dataset. However, we realized that tasks of this nature require significantly more computing power than we currently have access to.

The tasks were well-detailed and manageable to divide between two team members. It was convenient that the assignment was designed for two people, as a larger team (e.g., three or more members) might have introduced organizational challenges in dividing the workload effectively.

Overall, for future work and better performance, we would require more time and greater computing resources.

References

[1] MattD. [n. d.]. GitHub - MattD82/Predicting-Spotify-Song-Popularity: Repository for Data Science project. https://github.com/MattD82/Predicting-Spotify-Song-Popularity?tab=readme-ov-file

[2] Northdata. [n. d.]. Sony Music Entertainment Germany GmbH, Berlin. https://www.northdata.de/Sony%20Music%20Entertainment%20Germany%20GmbH,%20Berlin/Amtsgericht%20Charlottenburg%20%28Berlin%29%20HRB%20228359%20B

[3] Philip Peker. 2023. Predicting Popularity on Spotify — When Data Needs Culture More than Culture Needs Data. (9 2023). https://towardsdatascience.com/predicting-popularity-on-spotify-when-data-needs-culture-more-than-culture-needs-data-2ed3661f75f1

[4] Simon Tritscher and Tuvshin Selenge. n.d.. Spotify Analysis. https://github.com/TuvshinSelenge/spotify_analysis. Accessed: 2024-12-15.

A Additional Figures

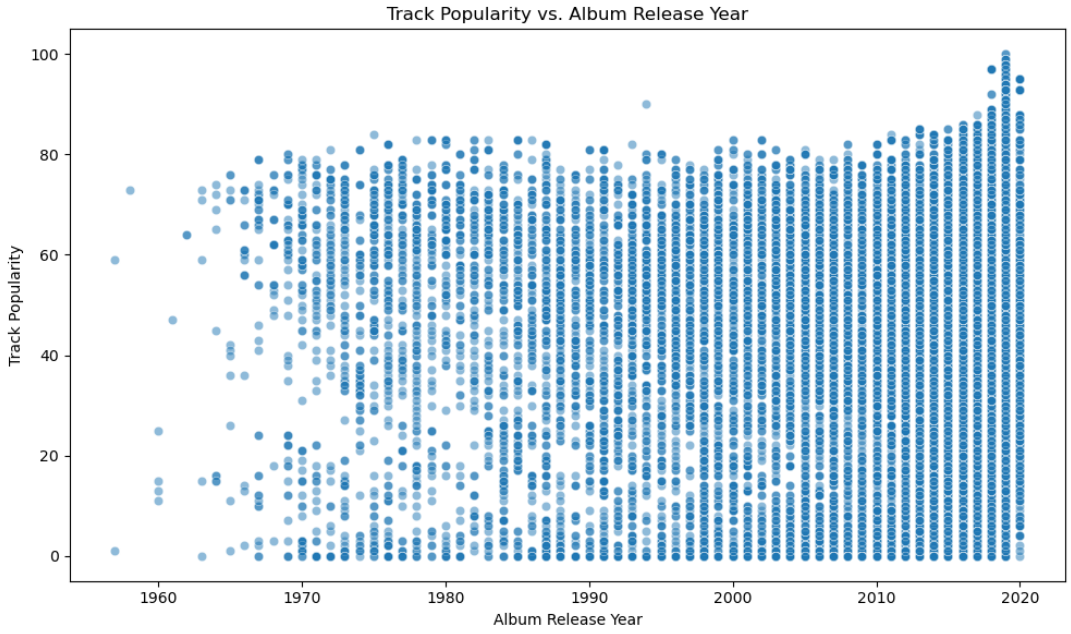

Figure 3: Popularity scores in comparison with album release year.

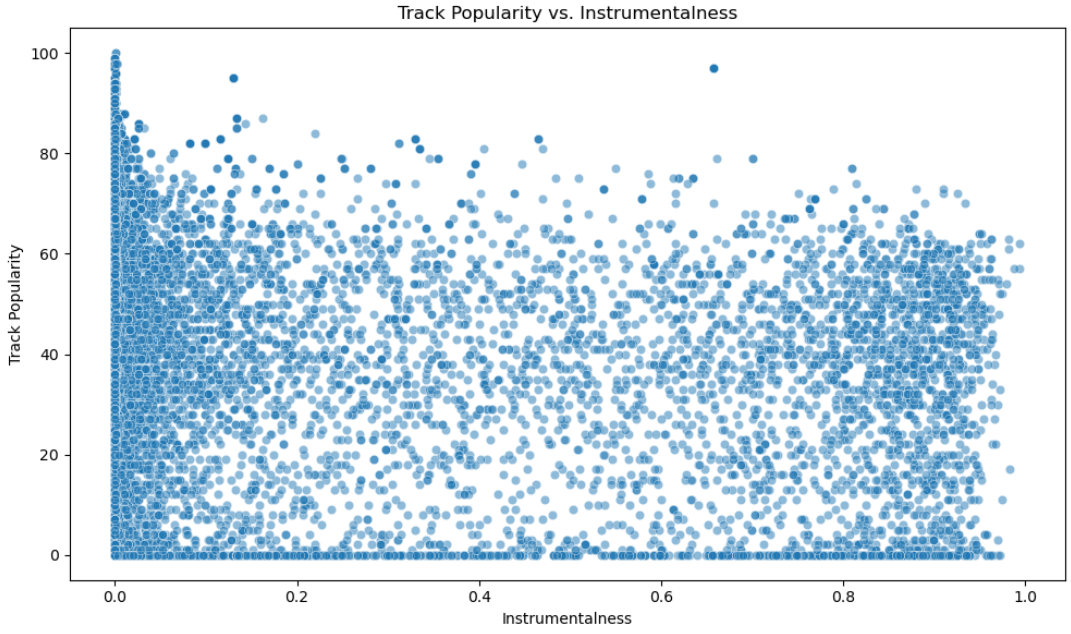

Figure 4: Popularity scores in comparison with instrumentalness.

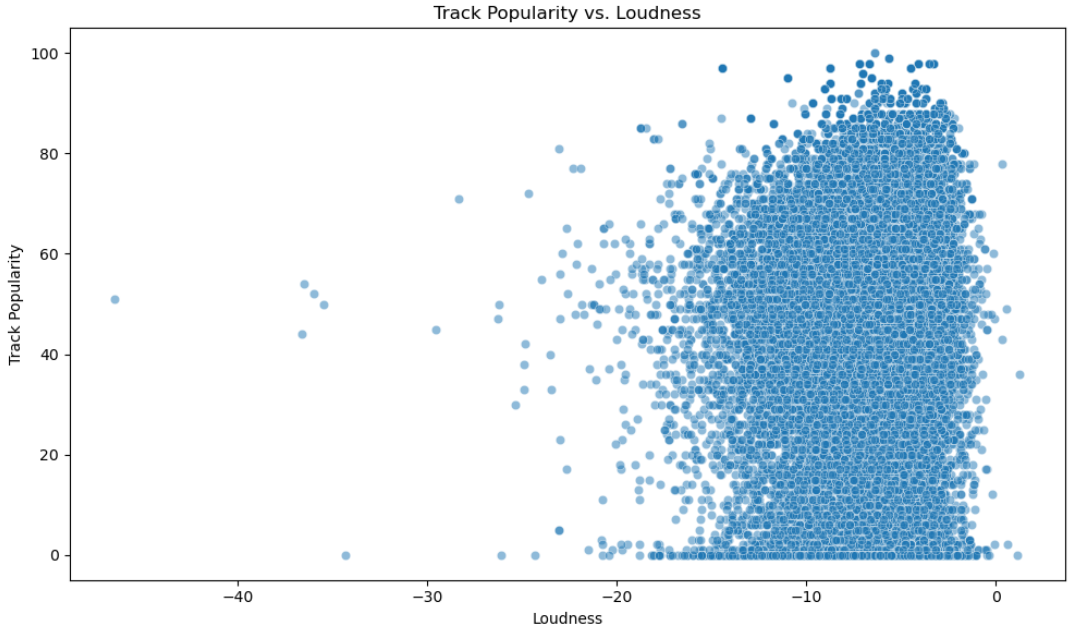

Figure 5: Popularity scores in comparison with loudness.

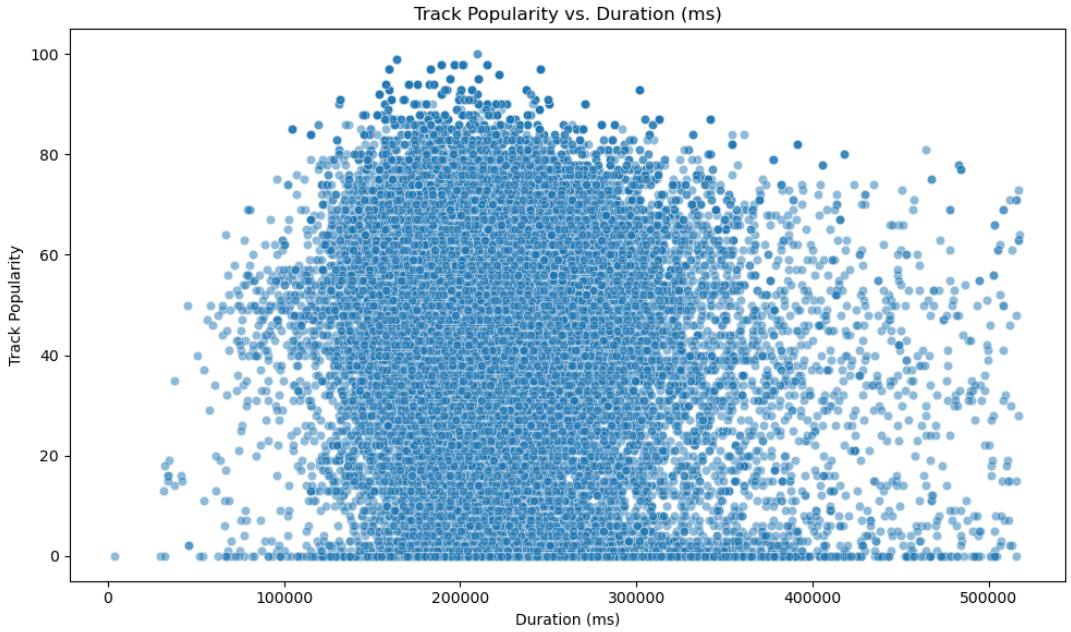

Figure 6: Popularity scores in comparison with track duration.

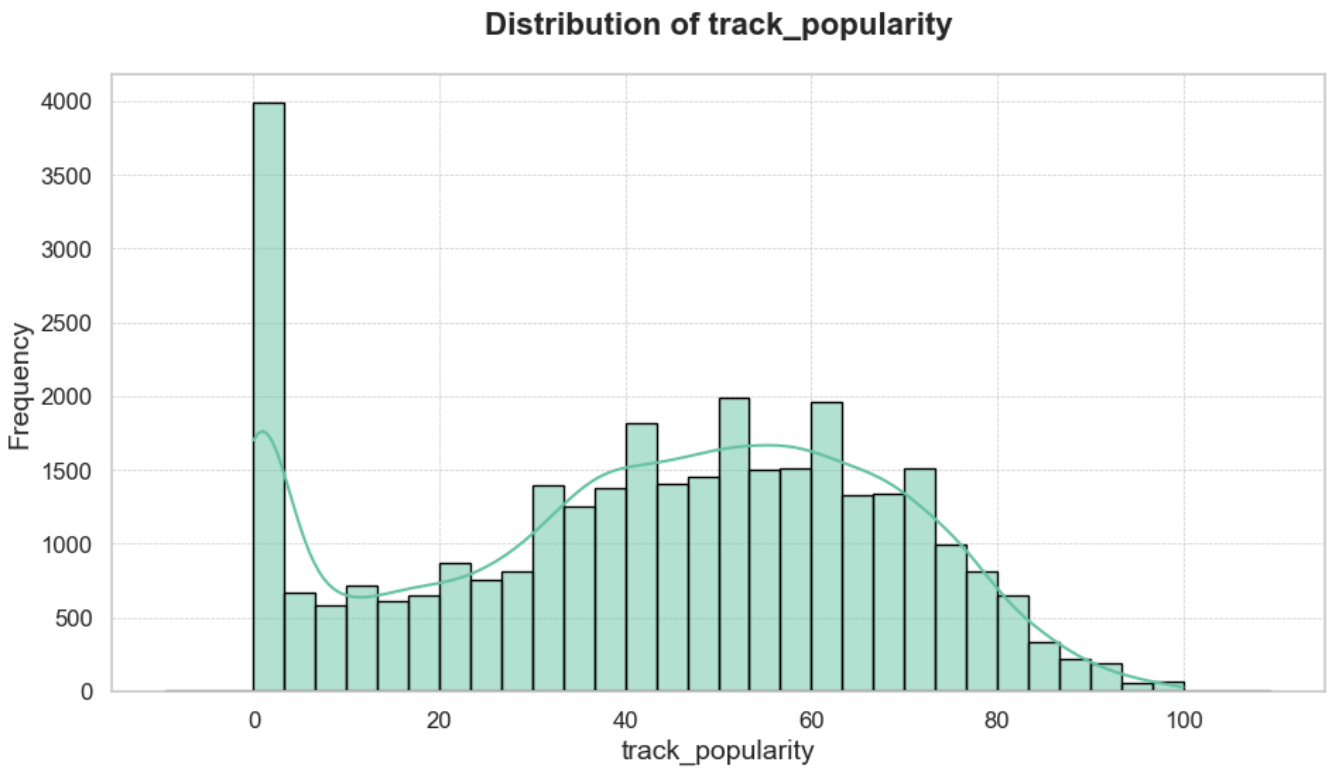

Figure 7: Popularity of the tracks.

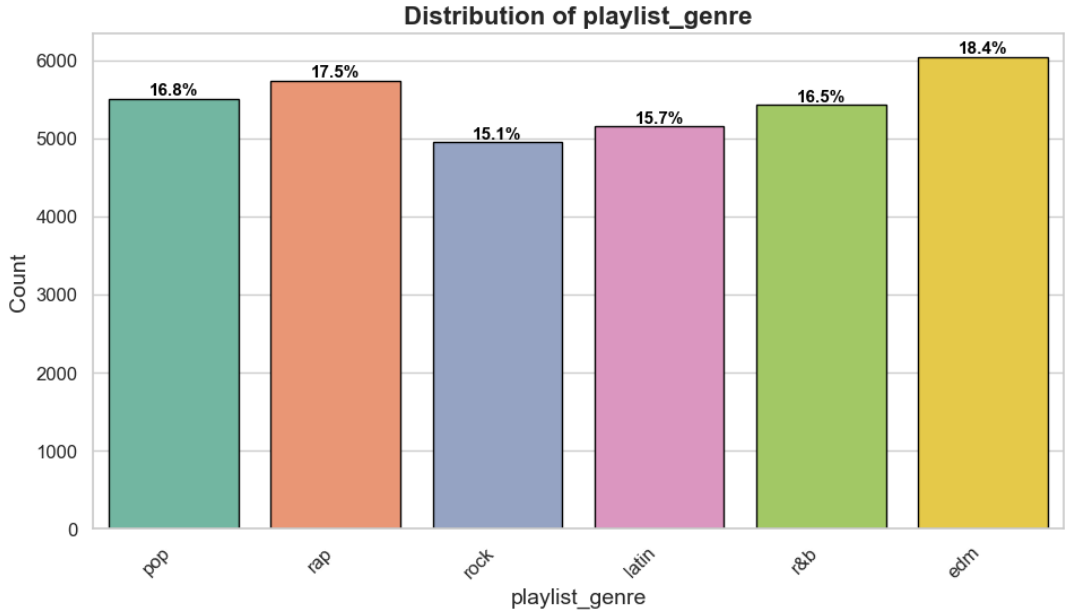

Figure 8: Genre of the tracks.

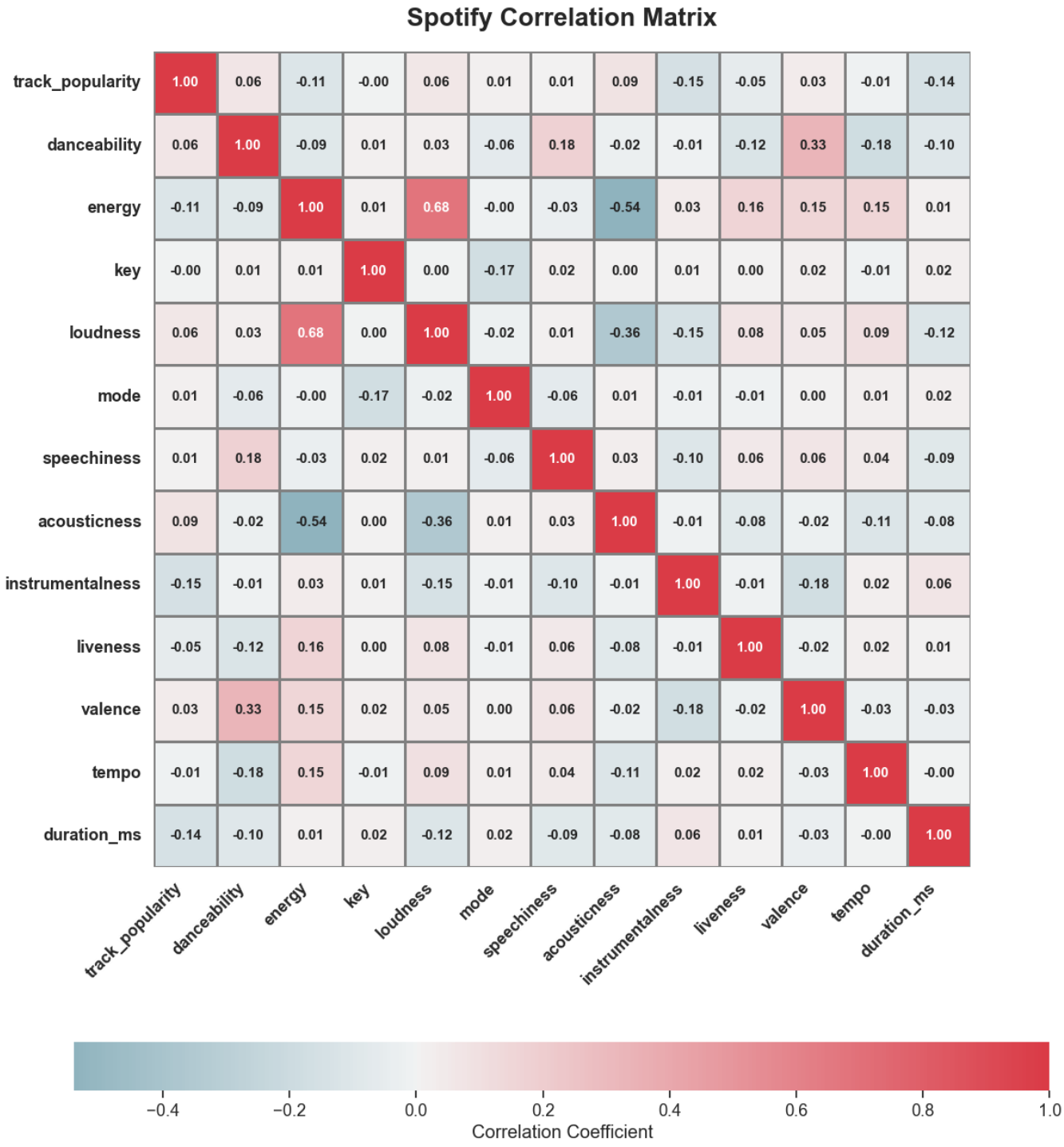

Figure 9: Correlation Matrix Heat-map