Analysis of Spotify's Audio Features Trends using Time Series Decomposition and Vector Autoregressive (VAR) Model

D A G Machmudin[1,*], M Novita[12], G Ardaneswari[3]

[1] Statistics Department, University of Indonesia, Depok, Jawa Barat 16424 Indonesia
[2] Actuarial Department, University of Indonesia, Depok, Jawa Barat 16424 Indonesia
[3] Mathematics Department, University of Indonesia, Depok, Jawa Barat 16424 Indonesia

*Corresponding author's e-mail: daffa.ghifari@gmail.com

Abstract. Streaming is the most popular music consumption method of the current times. As the biggest streaming platform based on subscriber number, Spotify stores miscellaneous information regarding the music in the platform, including audio features. Spotify's audio features are descriptions of songs features in form of variables such as danceability, duration, and tempo. These features are accessible via Application Programming Interface (API). On the other hand, Spotify also publishes their own charts consisting of 200 most streamed songs on the platform (based on regions) which are updated daily. By combining Spotify's song charts and the songs' respective audio features, this research conducted analysis on musical trends using time series modeling. First, the combined data is decomposed to extract the trend features. Second, a Vector Autoregressive (VAR) model is built and followed by forecasting of the audio features. Lastly, the performance of forecasted values and the actual observations is evaluated. As a result, this research has proven that musical trends can be forecasted in the future for a short period by using VAR model with relatively low error.

1. Introduction

Music is an immersive universal communication medium with inherent presence in human life [1]. Listening to music was once exclusive to direct activities such as ceremonies and concerts [2]. However, the condition has changed since the invention of the phonautograph, which can record and produce sound. The tool was invented by Edouard-Leon Scott de Martinville in 1857. After the presence of gramophone records, the format for listening to music underwent changes, starting from Walkman, CD, to digital files. These files can be distributed illegally and needs to be downloaded to listen to it made Daniel Ek and Martin Lorentzon discover Spotify, a streaming-based music service in 2006 [3]. Based on data from the Recording Industry Association of America, streaming has become the main music consumption format in the United States through various applications such as Spotify, YouTube, and Apple Music since 2016.

During the development of the music industry, there were various charts that act as useful parameters for music popularity, such as the Billboard Hot 100 charts and the Billboard 200 album charts. With the digitization of music, popularity can be inferred easier due to presence of streaming applications that provides the number of streams at frequent intensity. In addition, various music services also have other features that can describe music quantitatively, such as audio features on Spotify. Various studies regarding the popularity of songs in the music industry have been carried out with the help of streaming applications. Based on the audio features of Spotify, research conducted Kim and Oh [4] analyzed the songs features that have occupied the top 10 of Billboard Hot 100 charts with the said features. Kim[5]predicted the level of popularity of songs provided by Spotify based on their characteristics and compares them through three methods: linear regression, KNN regression, and Random Forest.

On music streaming platforms, there are two types of streams, namely streams of provided content (front-end streams) and streams in the form of user digital activity data (back-end streams) [6]. The data provided by the music platform can be seen as various forms including as time series. Time series data is formed to understand stochastic movements from observations and predict future values based on existing time series data [7] Time series data can be decomposed: breaking a model into several latent variables to better understand the characteristics of the data. After decomposition, time series data is divided into three parts, namely trend, seasonality, and residual[8]. As the main incentive of time series data is prediction, one of the suitable models for conducting time series analysis is the Autoregressive Integrated Moving Average (ARIMA) model. The multivariate form of the ARIMA model is often used on real data, namely the Vector Autoregressive (VAR) model [9]

Popularity analysis for a music industry landscape that continues to provide real-time data requires dynamic analysis. Based on various studies conducted, it has been proven that the elements of a song can support the popularity of that song. In the development of the music streaming era which provides real-time data in the form of a time series, the author intends to conduct research on song trends in the music industry based on the acoustic features provided by Spotify. The trend in question is the component resulting from time series decomposition of song data on Spotify in the form of weekly movements of audio features. Then, the extracted trend components will be analyzed using Vector Autoregressive (VAR) time series to predict the characteristics of music in the future based on the predicted audio features. This research aims to analyze song trends on Spotify while utilizing time series decomposition and VAR model.

2. Audio Features

The audio features on Spotify can be accessed using the Application Programming Interface (API) which acts as a way for two or more computer programs to communicate with each other. As the name implies, API is an interface software, meaning it provides services to parts of other software. Spotify Representational State Transfer (REST) API where programs can use it to retrieve and manage data for the music streaming service on the internet [10]. The protocol or rules for transmitting data used by the Spotify Web API is hypertext transfer protocol (HTTP) that happens to be the same as an internet browser. Hence, Spotify’s Web API can be accessed by the browsers of all internet users.

One of the endpoints of the Spotify Web API is the ability for developers to get information of a song from audio features. On the Spotify for Developers page, there are many audio features listed and four of which, that are used in this research, are:

- *Danceability*, describes a song’s suitability for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0 indicates a song is least suitable for dancing and 1 marks the opposite.
- *Duration*, states the duration of a song in milliseconds.
- *Tempo*, the overall estimated tempo of a song in beats per minute (BPM).
- *Valence*, a 0 to 1 measurement that describes musical positivity conveyed by a song. Songs closer to 1 tend to sound more positive (e.g. happy, cheerful, and euphoric) while the opposite also applies.

3. Data Collection

The data collection in this research was divided into two parts. The first part is collecting the most popular songs weekly via the official Spotify charts page while the second is collecting audio features via the Spotify Web API. The process carried out on the Spotify charts page aims to gather information regarding 200 songs that have managed to get the most streams on each weekly basis. The charts used are the weekly global charts curated by Spotify. On the second part, the process is conducted to retrieve data regarding each songs' audio features that have ever appeared on the chart. To assure that the correct audio features are assigned to the songs, these two stages have an important unique feature that a song has, which is called a uniform resource identifier (URI).

3.1. Data collection through Spotify's Global Weekly Chart

The weekly Spotify Global charts was downloaded manually from the chart's official website published by the streaming service from the first week of January 2017 to the third week of May 2023. Each weekly chart is downloaded and stored in a single file; thus this research has a total of 334 files. Various information was provided in each weekly file, some of which are the song title, number of streams sorted from highest to lowest, and URI. All files were then merged into a large database with 66.800 rows through the help of Python. Note that the number of songs displayed in this section is redundant as popular songs are very likely to occupy the charts for more than one week. This file will hereinafter be referred to as *the first file*.

3.2. Audio features collection

The first file from the previous section will be used to retrieve audio feature information via the Spotify Web API. The mechanism consists of instructing the API to search for audio features available on the Spotify server. Keep in mind *the first file*'s data size is considerably large with many repetitions. To simplify computing process, the feature collection step will only use Spotify's API with more concise data. A new dataset is formed with only the uniform resource identifier (URI) for each songs will be shown to make this succeeding dataset features unique songs. 5923 unique songs are obtained that have charted in Spotify's weekly charts with this dataset will be referred to as *the second file* hereinafter.

The next stage is to search for four audio features namely danceability, duration, tempo, and valence. This process is conducted by matching the URI of each song from *the second file* and the Spotify server endpoint. During the process, Spotify's Web API has a certain limit of requests to fetch audio features at a time. To avoid exceeding the limit, the Web API program is given an additional command in the form of a sleep process (stopped momentarily) with a random value according to a uniform distribution between 0 and 5 seconds. Furthermore, the data in the second file is divided into six separate data frames, each containing at most 1000 songs.

After the search is carried out, the audio features are stored in new columns added to each data frame. Then, the six data frames will be merged back into the initial form of *the second file*.

3.3. Pre-processing

Data Pre-processing in this study aim to improve the data format to suit this research's objectives: analyzing trends in songs that are currently popular on Spotify. Trend in the objective means the mathematical trend component (that can be obtained through decomposition) of the songs' audio features every week. Therefore, an aggregate process will be carried out to summarize the audio feature values by averaging each feature per 200 songs that results more simplified value of the audio features on a weekly basis. In addition, the audio feature summary will have a time series form that can be processed using decomposition methods and forecasting models.

Before summarizing, *the first* and *second files* from the data collection process need to be combined. Merging process is carried out to obtain *the first file* with complete audio features. The said file was beforehand mentioned to be redundant, yet this step must proceed to get complete audio feature values every week. Merged data from the first and second files or will be referred to in this study as *complete data*. The summarizing process will be conducted through averaging as an aggregate method in data pre-processing as it has an unbiased estimator property. *Complete data* that has been summarized has a size of 332. As a result of the summarization, not all columns of *the complete data* can be retained hence the only available information available are the date and the four audio features that will be studied further.

4. Time Series Decompostion

Time series decomposition method in this research is used to analyze a certain time series data's characteristic. The process may vary depending on the methods used, two of which are classical decomposition and seasonal-tren decomposition based on Loess. Before conducting any decomposition processes, time series data should be visualized to simply infer its characteristics.

4.1 Data Visualization

An important step in statistics that needs to be conducted between data pre-processing and data processing is data visualization. Data visualization aims to overview a data's characteristic. In this study, the data visualization stage was carried out to see the trend of audio features on the Spotify. The results of data visualization are shown as figure 1.

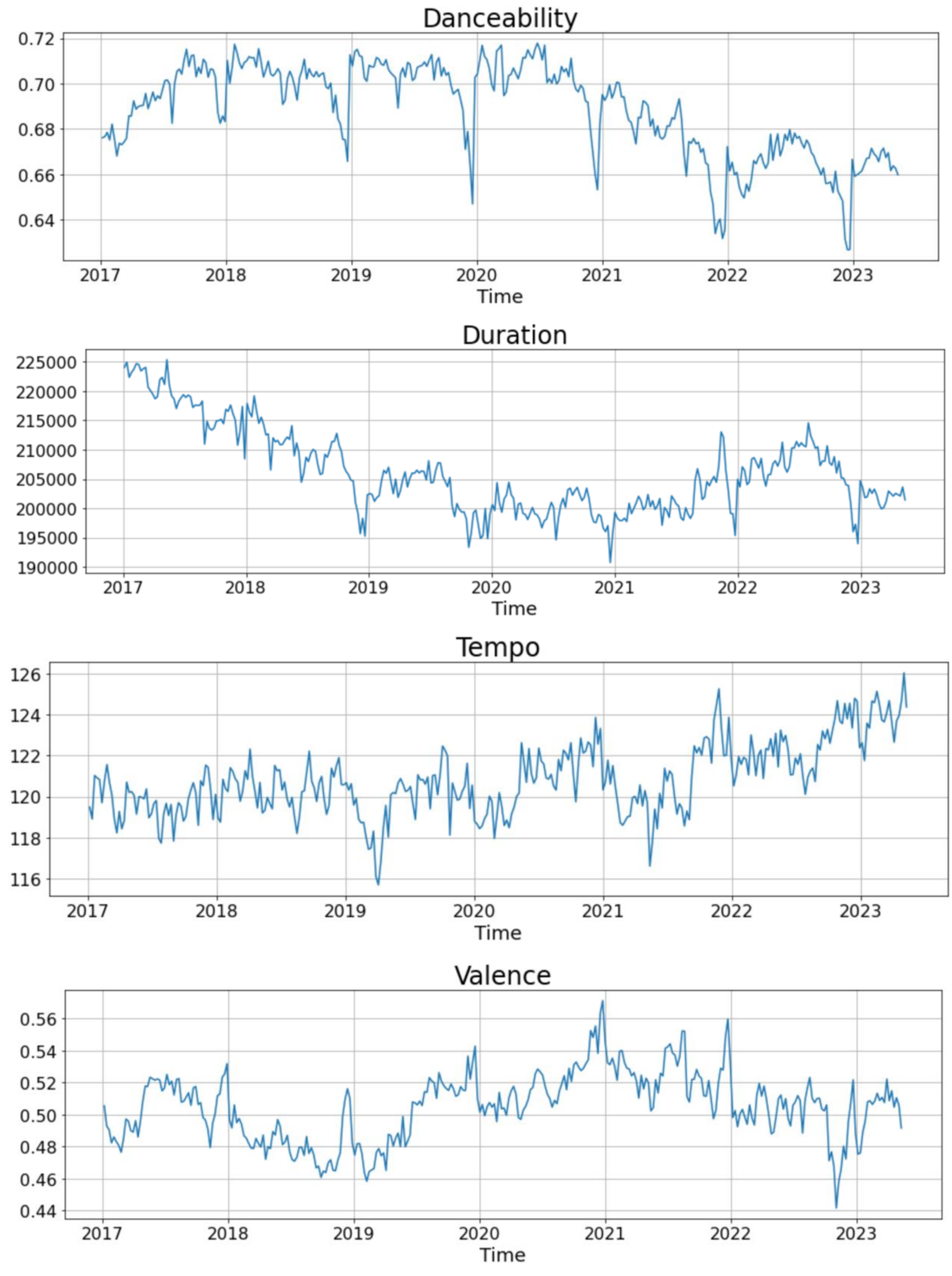


Figure 1. Visualization of the four audio features of *the complete data*.

In the danceability audio feature plot, the value of this audio feature is relatively high from 2017 to early 2021 and continues to slowly decrease until 2023. An exception occurred at the end of each year where a drastic decrease happened and directly followed by an increase to high values again in the first week of the new year. This might occur due to the evident of Christmas music culture. The duration feature at first glance experienced a constant decline from early 2017 to mid-2021, followed by an increase until it peaked in mid-2022, and back to decreasing until 2023. The tempo feature appears to have a constantly increasing trend. Lastly, valence has fluctuating movements.

4.2 Classical Decomposition

Classical decomposition is the most popular time series decomposition technique that uses moving averages method. One of the main reasons in using classical decomposition was due to its simple procedure. Broadly speaking, classical decomposition process consists of calculating the moving average (MA) value, detrending, and estimating the seasonality along with residual components.

To begin classical decomposition procedure, moving average method is used to estimate the cyclical-trend component. An MA of order m can be expressed as

$$\hat{T}_t = \frac{1}{m}\sum_{j=-k}^{k} y_{t+j} \tag{1}$$

where $m = 2k + 1$. Equation (1) is referred to as $m - MA$. The estimate of the cyclical-trend component at time t is obtained by averaging the values of the time series in k periods from t.

Classical decomposition can be divided into two categories, namely additive and multiplicative. This paper will only use the additive classical decomposition method although both categories may have the same assumption: the seasonal component is constant from year to year. The stages carried out in additive classical decomposition are calculating:

- Cyclical-trend component $\hat{T}_t$ using $m - MA$ for m odd numbers or $2 \times m - MA$ for m even numbers.
- Detrended series using $y_t - \hat{T}_t$.
- Seasonal component. There are several more steps to obtain the seasonality component.
  - First, average the detrended values for a pre-determined season. The seasonal component is obtained by calculating the average value for each season. For example, in monthly data, the seasonality component for March is the average of all detrended values from March each year in the data.
  - Second, ensure the sum of all values of this seasonal component (or unadjusted seasonality) is equal to zero. If it is not equal to zero, then adjustments need to be made by reducing the seasonal value obtained by the average of all seasonal components. The components that have been adjusted will be called adjusted seasonality.
  - Third, replicate the value of the seasonality component every year in the data, hence each value for each season is the same. Through these steps, the seasonal component will be obtained, namely $\hat{S}_t$.
- Residual component by subtracting the seasonal and trend-cyclical component values by means of $\hat{R}_t = y - \hat{T}_t - \hat{S}_t$.

Classical decomposition is commonly used due to its simplicity, although it is less recommended due to several problems that might occur during the method's usage. First, classical decomposition tends to over-smooth the cyclical-trend component for data with rapid increases or decreases. Second, this method assumes that the seasonal component repeats itself from year to year which hardly make sense (even though the assumption can be used over long time series). Third, this method is not robust for outlier values. Therefore, it is recommended to use other decomposition methods [11].

4.3Seasonal-Trend Based on Loess (STL) Decomposition Method

STL decomposition in this study is referred to prior research conducted by Cleveland, et. al. [12]. STL decomposition consists of two recursive processes, an inner loop and an outer loop. The inner loop component is an iterative process to calculate trend and seasonality components, while the outer loop is conducted to reduce the influence of observations with different characteristics (outliers). An iterative process of each loop is referred to as a pass. Additionally, in STL there will be several parameters that will be explained further in this part.

Inner loop process is carried out $n_{(i)}$ times, while the outer loop process is carried out $n_{(o)}$ times. For each *pass* in inner loop, seasonal and trend components will be updated once whereas in the outer loop, each *pass* consists of an inner loop followed by computing robust weights. The condition for the first pass in the outer loop is that the robust weight has a value of 1.

Another parameter in STL decomposition is $n_{(p)}$ that states the number of observations from a period. For example, a monthly time series that has annual periods will have a value of $n_{(p)}$ of 12. Given the magnitude of each period, the values from the first to the last period have a certain order. From each value in the same order a sub-series can be formed which is called a Cycle-subseries. For example, in monthly data with $n_{(p)}$ of 12, the Cycle-subseries of the first order is all January values.

Each inner loop pass is carried out to earn seasonality and trend components through loess smoothing. The two said components are defined as $S_t^{(k)}$ and $T_t^{(k)}$ for $t = 1,2,\dots,n$ where k represents the number of passes that is conducted. These components are defined at every point even though Y_t does not exist value. Updates after the pass in the $(k+1)$ stage for $S_t^{(k+1)}$ and $T_t^{(k+1)}$ are carried out according to the steps below, although generally the inner loop process is only done once. Note that at the STL decomposition stage there are parameters $n_{(l)}, n_{(s)}, n_{(t)}, n_{(o)}, n_{(i)}$, and $n_{(p)}$whose value selection will be discussed further in the study. The steps involved in the inner loop are:

- Detrending by counting $Y_t - T_t^{(k)}$. If Y_t doesn't exist, the *detrended* series is nonexistent. As an initiation the value of $T_t^{(0)}$ is 0.
- Smoothing at the cycle-subseries level using Loess with $q = n_{(s)}$ and $d = 1$ for each cycle-subseries. Calculations are performed at all points from just before the first point to right after the last point of the cycle-subseries. All smoothing results will form a seasonal temporary series $C_t^{(k+1)}$ consisting of $N + 2n_{(p)}$ values ranging from $t = -n_{(p)} + 1$ to $N + n_{(p)}$ where N denotes the number of observations.
- Performs low-pass filtering of cycle-subseries. The term low-pass filter implies the filter ignores high frequency values while allowing low frequency values to pass through. This filter consists of two moving averages with length $n_{(p)}$, followed by a moving average with length 3, and a smoothing loess with $d = 1$ and $q = n_{(l)}$. This low-pass filter will be applied to $C_t^{(k+1)}$ and give the output $L_t^{(k+1)}$ which is defined at $t = 1,2,\dots,N$.
- Calculating the seasonal component by detrending from the smoothed cycle-subseries. The seasonal component of the $(k+1)$ to loop is $S_t^{(k+1)} = C_t^{(k+1)} - L_t^{(k+1)}$ for $t = 1,2,\dots,N$.
- Deseasonalizing by calculating $Y_t - S_t^{(k+1)}$. If the Y_t value does not exist, then the deseasonalized value does not exist either.
- Performs smoothing of trend components. The series that has been deseasonalized is then smoothed with loess with $q = n_{(t)}$ and $d = 1$ at $t = 1,2,\dots,N$. The result of this smoothing is the trend component $T_t^{(k+1)}$ for $t = 1,2,\dots,N$.

Based on the stages above, the seasonal-smoothing portion at stages 2, 3, and 4 while the trend-smoothing stage is at stage 6. After the inner loop process, the trend and seasonal components estimation values are obtained, which formed sequences T_t and S_t respectively.

Subsequently following inner loop process, it is necessary to perform the outer loop process in effort of reducing the residual component's influence on the time series data. First, residual component is defined as R_v which can be calculated exactly like classical decomposition's residual: $R_t = Y_t - T_t - S_t$.

An important note for Y_t is the missing values are defined. Then, a robust weight is defined for each point Y_t to find out how extreme the value of R_t is. An outlier with a value of $|R_t|$ which is large will have a small weight or close to zero. For example, $h = 6\,median\,(|R_t|)$ is defined, then the robust weight of point v is $\rho_t = B(|R_t|/h\,)$ where B is the bisquare weight function, which is

$$B(u) = \begin{cases} (1-u^2)^2\ , & 0 \le u < 1 \\ 0\ , & u > 1. \end{cases}$$

Once the weights are determined, the inner loop stage is carried out again but only at the second and sixth smoothing stages, the neighborhood weight value of a value at time t is multiplied by the robust weight ρ_t. The robust iteration of the outer loop is performed $n_{(o)}$ times. At each stage of the inner loop after the initial pass, $T_t^{(0)} = 0$ is no longer used but instead uses the trend component from stage 6 of the previous inner loop.

With the loops being done, a post-smoothing procedure can be conducted to produce a smoother seasonal component. For example, a seasonal component may change smoothly from one year to the next but less so from one day to the next. Therefore, seasonality component can be smoothed with loess.

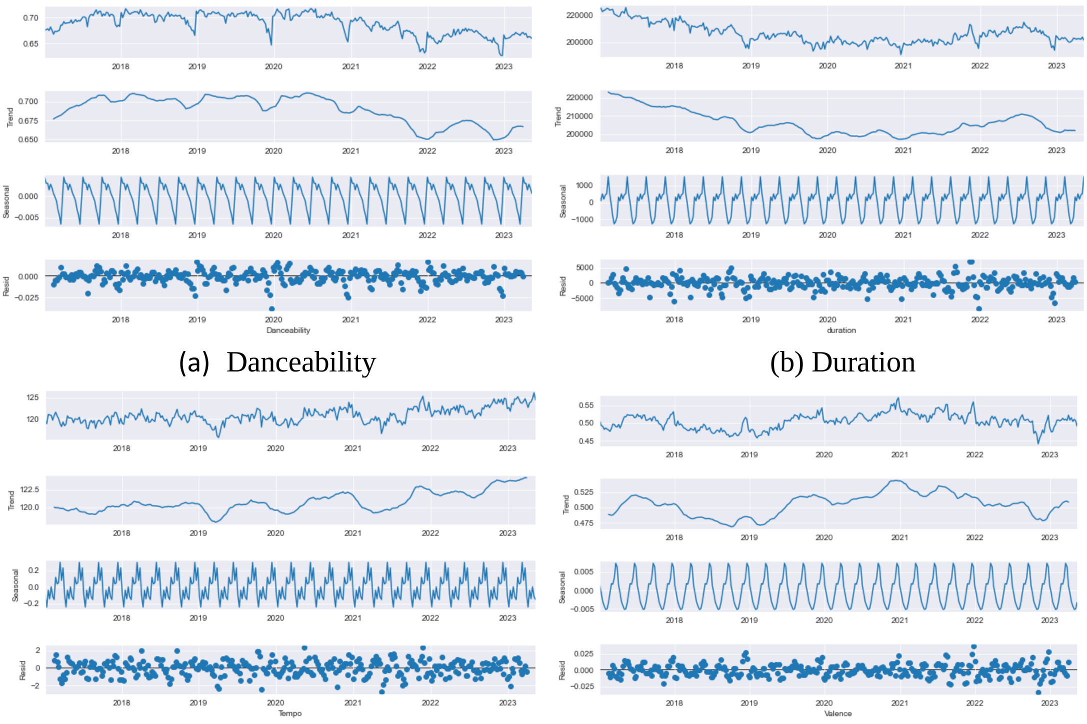
In STL decomposition, there are six parameters that needs to be estimated:

- $n_{(p)}$ = number of observations in a cycle,
- $n_{(i)}$ = number of passes through *inner loop*,
- $n_{(o)}$ = numbers of *robust* iterations at *outer loop*,
- $n_{(l)}$ = *smoothing* parameter for *low-pass filter*,
- $n_{(t)}$ = *smoothing* parameter for trend component,
- $n_{(s)}$ = *smoothing* parameter for seasonal component.

4.4 Comparisons of Audio Features Decomposition

The main objective of decomposition in this research is to extract trend components from time series regarding audio features. In this study, the extraction of the trend component came from STL decomposition results, while the classical decomposition results were only used as a comparison of the former method. It should be noted that the decomposition process is carried out univariately for each audio feature.

Additive classical decomposition will be used for the four variables in this research: danceability, duration, tempo, and valence. In classical decomposition which uses a moving average, a parameter m is required which stands for period. The periods chosen in this study is 13 due to the calculation of 13 weeks equivalent to one quarter. Said period came into consideration and was later determined since countries that contribute to the Spotify chart more often have a four-season climate with one season lasting 13 weeks. Additive classical decomposition of the four audio feature components is shown in accordance with Figure 2.



(a) Danceability (b) Duration

(c) Tempo (d) Valence

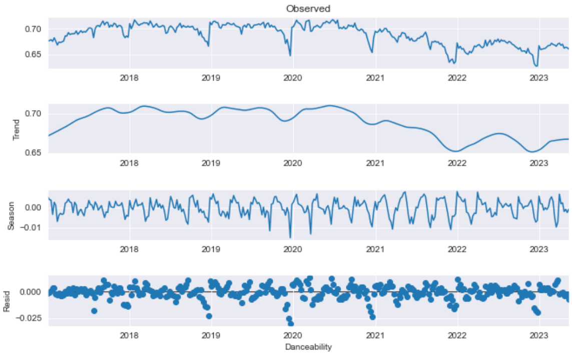
Figure 2. Classical decomposition of the four audio features.

The decomposition process shows that trend components have various patterns in the four audio features. In the danceability, the trend component was relatively stable at 0.70 from the first week to the 200th week and decreased thereafter. Duration's trend component showed decrease since the beginning of 2017 until the 100th week of the study and tends to be stable until the end. The tempo trend component had constant increment from early 2017 to 2023. In the valence component, the trend component formed waves with valleys and hills that occur sequentially at the end of 2018 and early 2021.

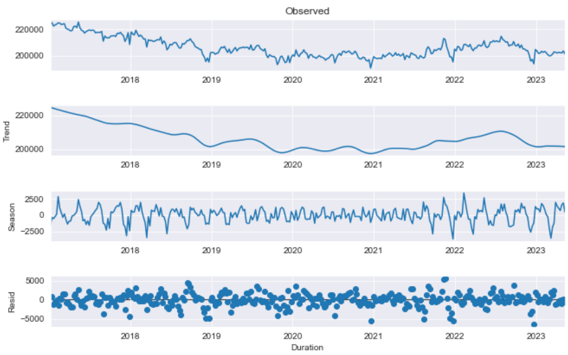
As a rule of decomposition, it was necessary to determine several parameter values. However, as this research is conducted through Python, the program only had one parameter set $n_{(p)}$ (number of observations set for one season period). This can occur due to calculations from the program that can determine other parameters based on applicable rules or $n_{(p)}$. The provisions in question are:

- $n_{(i)}$ and $n_{(o)}$ are decided depending on *robustness* for observations with outlier.
- $n_{(s)}$ is an odd integer bigger than or equal to 7.
- $n_{(l)}$ is an odd integer that is bigger than, or equal to $n_{(p)}$.
- $n_{(t)}$ depend on a calculation that will not be discussed in this paper.

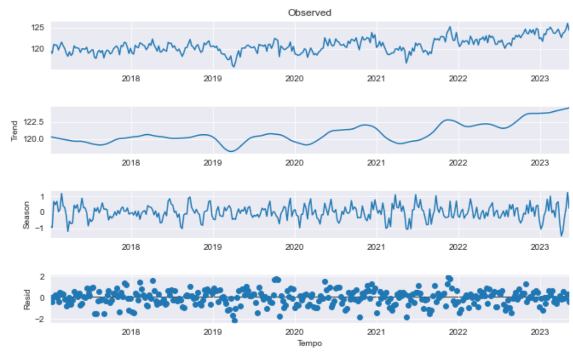
The selection of the period parameter $n_{(p)}$ is set at 13 for reasons similar to the classical decomposition regarding significant changes in value every year. As a result of choosing the value of $n_{(p)}$, the values of other parameters according to the provisions are 7 for $n_{(s)}$, 13 for $n_{(l)}$, and 25 for $n_{(t)}$. Selected $n_{(i)}$ and $n_{(o)}$ values are 1 and 0 due to the absence of significant outliers in the four audio features. STL decomposition on the danceability, duration, tempo, and valence components are shown in Figure 3.



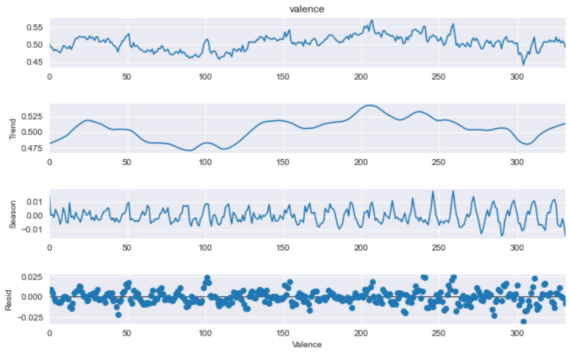
(a) Danceability



(b) Duration



(c) Tempo



(d) Valence

Figure 3. STL decomposition of the four audio features.

Overall, the results of the STL decomposition show similarities to the additive classical decomposition with some notable differences due to the nonparametric nature of the STL decomposition. However, the trend component of the STL decomposition results has a more volatile form. This can occur due to the use of the nonparametric Loess function. The seasonal component also has a pattern similar to the classical decomposition but there are fluctuations in its values.

In the seasonal component, there are extreme values at the turn of the year for all audio features except tempo. The audio tempo feature has a seasonal component with hills that appear periodically only since the 50th week. In the danceability and duration features, there is a sharp decrease at the turn of the year but is followed by a value that returns to its original level. Valence shows peaks that occur periodically every year.

5. Vector Autoregressive (VAR)

Vector Autoregressive model may only be formed under several assumptions. If the conditions are not met, presumably another model would be a better fit for modeling the data. In this research, there are three assumptions that will be tested. Subsequently, data modeling was carried out using the VAR model and predictions were made for the next few weeks.

5.1 Assumptions

There are underlying 3 assumptions that need to be tested chronologically to build autoregressive vector model: Granger Causality, Stationarity, and Cointegration. First, granger causality test is conducted then a following stationarity test is performed. If the data happens to be non-stationary, it is necessary to carry out a cointegration test. In case cointegration exists, the data will be more suitable to be modeled using vector error correction model (VECM). In turns of no cointegration, stationary differencing is carried out until stationary data is obtained for modeling.

5.1.1 Granger Causality. In a regression analysis, the process is carried out to see the relationship between explanatory variables and response variables in a data. The relationship of the two variables does not fully indicate causality or direction of the relationship. However, regression on time series data may have misleading interpretations as time does not run backwards.

For example, a variable A is declared to cause Granger causality (Granger-cause) of another variable B if the values of B and A in the past have better predictive ability than the value of B in the past alone. If it turns out that variable B does not Granger-cause variable A, then variable B is *strictly exogenous* variable to variable A. Exogenous variable means the variable that cannot be explained by other variables in a model. The Granger Causality test can be carried out using Box-Jenkins method with ARIMA model or multiple regression model with ordinary least squares or OLS method.

In this study, OLS method will be used to compare the unrestricted and restricted model. The hypothesis for Granger Causality test is:

$H_0 : \forall \beta_i = 0$ or Granger *causality* is inevident

$H_1 : \exists \beta_i \neq 0$ or X does Granger cause Y.

The above hypothesis needs to be tested using global F test according to the regression with the OLS method. To perform the test, a restricted DL model needs to be defined as in equation (3.16).

$$A_t = \alpha + \delta_t + \phi_1 A_{t-1} + \phi_2 A_{t-2} + \phi_3 A_{t-3} + \phi_4 A_{t-4} + \epsilon_t. \tag{3.1}$$

The unrestricted *Distributed Lag* (DL) with order (p, q) has form of:

$$\begin{gathered} A_t = \alpha + \delta_t + \phi_1 A_{t-1} + \cdots + \phi_4 A_{t-4} + \beta_1 B_{t-1} + \cdots + \beta_4 B_{t-4} + \\ + \psi_1 C_1 + \cdots + \psi_4 C_4 + \theta_1 D_1 + \cdots + \theta_4 D_4 + \epsilon_t \end{gathered} \tag{3.2}$$

where:

- A_t states *danceability* variable
- B_t states *duration* variable
- C_t states tempo variable
- D_t states *valence* variable
- $\phi_i, \beta_j, \psi_k, \theta_l$ are the parameters for each variable.

The two forms of models will be compared using global F test. Based on the results of the global F test, variable $(q, T - q - (p + 2))$ or p −value is less than the significance level α. In this study, the Granger Causality test was carried out for each variable with the results obtained were in the form of a p −value matrix for each comparison test between restricted and unrestricted models. Using a confidence level of 5% the results of the Granger Causality test are:

Table 1. Granger Causality Test (GCT) results.

	Danceability_x	Duration_x	Tempo_x	Valence_x
Danceability_y	1	0,0000	0,0000	0,0000
Duration_y	0,0000	1	0,0000	0,0000
Tempo_y	0,0000	0,0000	1	0,0754
Valence_y	0,0004	0,0175	0,0264	1

Based on the results, duration, tempo and valence variables had Granger cause on the danceability variable while danceability, tempo and valence variables had Granger cause on the duration variable. Then, danceability and duration variables were Granger caused to tempo variable. It can be inferred that valence variable is not caused by Granger cause from other variables. Since there are many possible variables that have Granger causes, the data can be continued to form a VAR model.

5.1.2 Stationarity. In stochastic processes, the term *stationary* implies a process is invariant to time. Mathematically, a stochastic process y_t is said to be stationary if the process mean value is constant for every point of time, and has an autocovariance value that only depends on the lag. In VAR, a stable VAR(p) process is stationary for $t = 0, \pm 1, \pm 2, \dots$. It should be noted that the converse of this stationarity condition is wrong as a stationary process is not necessarily stable.

In its application, stability is a mere underlying assumption. On the other hand, stationarity condition of a multivariate data must be tested. The stationarity test of the data to be formed by the VAR model can be carried out using the Augmented Dicky-Fuller (ADF) test in a unique manner like the AR model. However, if the data is found to be not stationary, a cointegration test needs to be carried out. Cointegrated data would be more suitable to be modeled with the Vector Error Correction Model (VECM). If there is no cointegration and nonstationary, each variable can be differentiated until it meets stationary conditions.

The most widely used stationarity test is the univariate Augmented Dicky-Fuller (ADF) test. For VAR's case even though ADF test is carried out individually, each component will have the same test hypothesis:

$H_0: \tau = 0$ or stationarity is not evident

$H_0: |\tau| < 0$ or stationarity is evident.

In this study, it is assumed that there is no stationary time series of trends in the four audio features. This may occur as the values used in the research are the trend component from decomposition results. Therefore, a cointegration test is carried out in the next subchapter. It was later found that there was no cointegration hence another differencing was conducted along with another round of stationary test. After differencing the data once, it was found that the data was stationary.

5.1.3 Cointegration. Data characteristics that need to be considered before building a VAR model are cointegration conditions. Cointegration is a situation when two or more nonstationary time series formed a stationary series though linear combination. According to Engle and Granger, cointegrated variables will have long-term stability (long-run equilibrium) but may experience short-term volatility (disequilibrium). In other words, cointegrated time series will move in the same direction, resulting in a stationary time series.

Cointegration can be explained using a mathematical model. Suppose a time series vector Y_t which consists of two different nonstationary time series:

$$y_{1t} = (y_{11}, y_{12}, \dots, y_{1t})$$
$$y_{2t} = (y_{21}, y_{22}, \dots, y_{2t}).$$

In the case of cointegration exist between the two, it can be shown the two time series are non-stationary yet may form a stationary linear combination series according to equation number (3.18)

$$\beta Y_t = \beta_1 y_{1t} + \beta_2 y_{2t} + \beta_2 y_{2t} \sim I(0) \tag{3.3}$$

In cointegration's context, β in equation (3.18) can be referred to as a cointegrating vector which is not necessarily unique. If a time series data has cointegration, said data will be more suitable to be modeled using an Error Correction Model (ECM) or in the case of a multivariate, Vector Error Correction Model (VECM). The VECM model can be estimated using Johansen method. The method can also determine whether time series data has cointegration or not.

There are two Johansen methods in the VECM context, the trace test and the maximum eigenvalue. Johansen test's main principle notes the long-term matrix Π will determine whether the variables in VAR(p) are cointegrated or not. In this research, the trace test is used with the hypothesis:

H_0: Cointegration rank of *unrestricted* VECM is $k = m$

H_1: Cointegration rank of *restricted* VECM is $k + 1$.

Johansen test uses likelihood ratio (LR) as its test statistic which is constructed from the diagonal values of the matrix of generalized eigenvalues for Π. If the LR statistical value is close to 0, H_0 is less likely to be rejected. But if LR is greater than the coefficient of variance then the initial hypothesis will be rejected.

On this research's test results, no cointegration in the vector of the four audio features was found. This condition implied the four audio features have time series movements in different directions. Mathematically, the four audio features do not move according to a linear combination that forms stationary time series. Therefore, the VAR model will be used to predict the four audio feature trends of the Spotify streaming service.

5.2 Modeling

The Vector Autoregressive (VAR) model is a multivariate form of the Autoregressive (AR) model. Even though it is analogously similar to AR, there are a number of things needs to be considered as in the AR model, modeling can only be done for one variable. The VAR model is often used in econometrics as it is seen suitable for modeling multivariate time series data. The form of the VAR model with the number of lags or order p (VAR(p)) is according to the equation (2).

$$y_t = v + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + u_t \qquad (4)$$

where:

- $y_t = (y_{1t}, \dots, y_{Kt})'$ is a rendom vector $(K \times 1)$
- A_i for $i = 1,2,3, \dots, K$ is a parameter coefficient matrix of VAR sized $(K \times K)$
- $v = (v_1, \dots, v_t)'$ is a constant vector $(K \times 1)$ yang which is an *intercept* to avoid zero mean
- $u_t = (u_1, \dots, u_t)'$ is white noise with K number of dimension.

VAR(p) can also be represented by VAR(1) which is as (3)

$$Y_t = \mathbf{V} + \mathbf{A} Y_{t-1} + u_t \qquad (5)$$

with

$$\mathbf{A} = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I_k & 0 & \dots & 0 & 0 \\ 0 & I_k & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_k & 0 \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}, \mathbf{V} = \begin{bmatrix} v \\ 0 \\ \vdots \\ 0 \end{bmatrix}, dan\ U_t = \begin{bmatrix} u_t \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where Y, V, and U are column vectors with kp dimension, and A is a $(kp \times kp)$ matrix.

Before the modeling stage, it should be noted that this research splits the data into two parts, namely training and testing in which the latter had 8 data points while the former had the rest of the data. Previously, this research's data had been qualified to be modeled by VAR through tests regarding the model's underlying assumptions. The first stage in building a model is specifying the model itself. This stage is particularly important to figure out the VAR model's order. In specifying and determining a model's order (lag), information criterion can be used. Akaike Information Criterion (AIC), Schwarz's Bayesian Information Criterion (BIC), and Hannan and Quinn Information Criterion (HQIC) are used in this research. From the criterions, the optimal number of lag order is 4. As a result, the VAR(4) model was determined to be used, which has the form:

$$y_t = v + A_1 y_{t-1} + A_2 y_{t-2} + A_3 y_{t-3} + A_4 y_{t-4} + u_t.$$

The parameters were later estimated using *ordinary least squares* (OLS) method before proceeding to the prediction stage.

5.3 Prediction

Arguably, the main objective of time series modeling is forecasting or predicting data in the future. Forecasting on the VAR(4) model to predict audio feature trends in Spotify is an extrapolation process, which means predicting data points outside the training data from the model. Ihas been explained that the data is divided into training data to build models and extrapolation and data testing to compare prediction results and observation results. After building the model, predictions were made of 8 data points on the four audio features. Comparison of the extrapolated results and the original values from the observations of the four audio features can be more easily interpreted using visualization. Therefore, the comparison plot of the testing data and the extrapolation results is in accordance with Figure 4. Note that the label with the element "_forecast" represents the extrapolation result of the VAR(4) model.

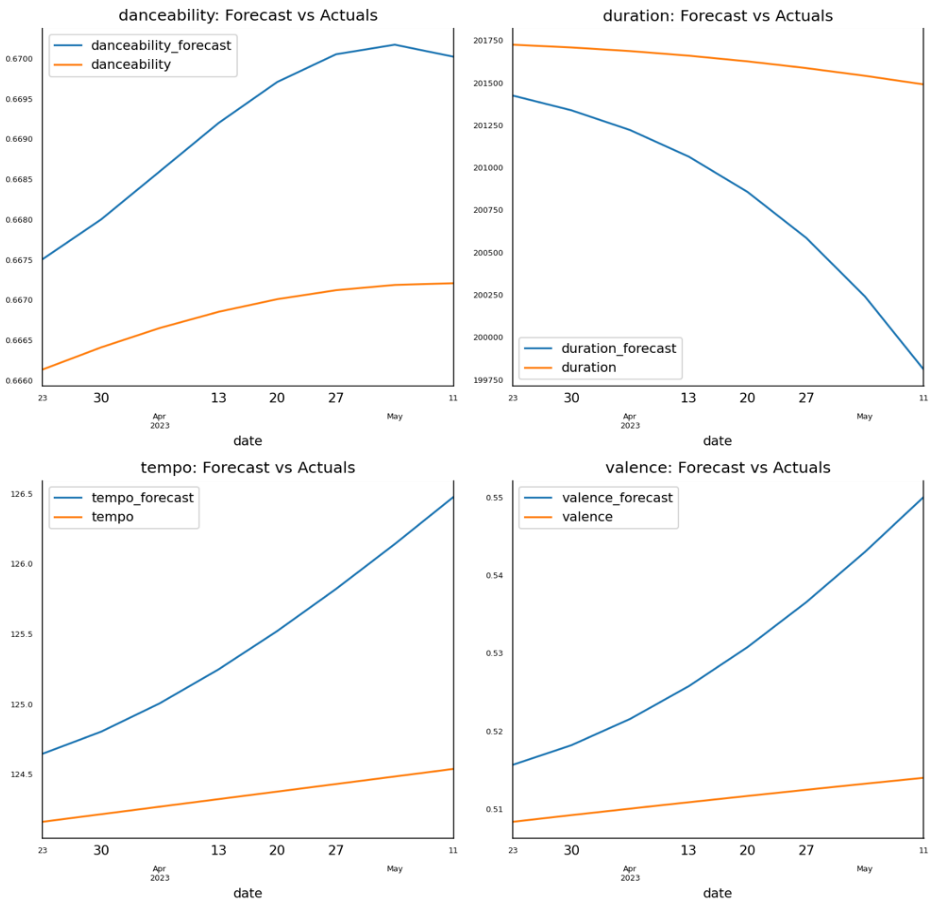


Figure 4. Comparison of forecast data from VAR(4) and actual data.

The comparison result of the predicted value and the actual value of the VAR(4) model shows that there is a difference between the two. In the duration feature, there is a downward trend in the predicted results of the VAR(4) model with consistently lower predicted values compared to actual values. In the danceability, tempo, and valence features, an upward trend existed with all predicted values having a higher value than the actual value. For the four audio features, the farther the predicted value is from the last data used to build the model, the faster the rate of increase/decrease is observed and away from the actual value.

Visually, there is a clear difference between the predicted values and the original observed values. To clarify the purpose of the comparison, the error level of the VAR(4) model will be evaluated. Accuracy level measurements are used using four measurements, namely mean absolute error (MAE), rooted mean squared error (RMSE), and mean absolute percentage error (MAPE). The results of the four forecast measurements are in accordance with table 2.

Table 2. Evaluation metrics of forecast results measurements

Fitur Audio	MAE	RMSE	MAPE
Danceability	0,0023	0,0029	0,0035
Duration	810,5624	929,6397	0,004
Tempo	1,1075	1,21	0,0089
Valence	0,019	0,0213	0,037

Based on the error measurement results from the VAR(3) model, it is evident that the prediction results from audio features have a very small error using MAPE measurements. Specifically, the MAPE values of the danceability, duration, tempo, and valence features are 0.35%, 0.4%, 0.89%, and 3.7% respectively. These results indicate that the model has a very good level of accuracy for danceability, duration, tempo, and valence features.

For relative model evaluation criteria, the results of the RMSE or MAE are as shown on table 2, but for this study it will be interpreted for RMSE only. In the danceability feature, the difference in value is 0.0029 units using RMSE. The duration feature shows the error value of the model is 929.6397 milliseconds or around 0.93 seconds. The error rate for the tempo is 1.21 beats per minute (BPM) and the valence is 0.0213 units.

6. Conclusions

Based on the discussion of the time series decomposition application used to build a Vector Autoregressive (VAR) model from the audio feature data of the Spotify music streaming service, the following conclusions can be drawn.

- The trend components of the audio features danceability, duration, tempo, and valence show different characteristics both in terms of movement from week to week and the direction of the trend.
- The STL decomposition process successfully extracts trend components with patterns similar to classical additive decomposition. However, the seasonal component resulting from STL decomposition has fluctuations compared to the seasonal component of classical decomposition due to the non-parametric nature of the Loess function.
- Trend forecasting for the four audio features is carried out using the VAR(4) model because it has passed the assumption test. The results of the model predictions show a visual difference between the predicted results and the actual values of the four audio features. However, the error level produced by the model is small for all audio features consisting of danceability, duration, tempo, and valence which respectively have error values of 0.35%, 0.4%, 0.89%, and 3.7 % according to MAPE.

Things that can be considered as input for further research regarding the application of time series decomposition in the VAR model using the Spotify audio feature are as follows.

- Charts are taken daily and differentiated for each country to obtain more varied sample characteristics. This needs to be done to increase the model's ability to learn changes due to shocks that occur more frequently in audio feature trends.
- Use more audio features to find out more information about the characteristics of trending songs that will be popular.
- Using other decomposition methods to extract trend components or using other models to predict audio features under certain conditions, such as using the Vector Error Correction Model (VECM) for data that has cointegration.
- Genre analysis of music uses other methods to be able to group trends in songs that are currently popular more generally.

References

[1] S. A. Mehr *et al.*, “Universality and diversity in human song,” *Science (1979)*, vol. 366, no. 6468, Nov. 2019, doi: 10.1126/SCIENCE.AAX0868/SUPPL_FILE/AAX0868_MEHR_SM.PDF.

[2] R. Philip, “Performing music in the age of recording,” p. 293, 2004.

[3] D. Arditi, *iTake-over : the recording industry in the digital era*. 2014.

[4] S. T. Kim and J. H. Oh, “Music intelligence: Granular data and prediction of top ten hit songs,” *Decis Support Syst*, vol. 145, Jun. 2021, doi: 10.1016/j.dss.2021.113535.

[5] J. Kim, “Music Popularity Prediction Through Data analysis of Music’s Characteristics,” *International Journal of Science, Technology and Society*, vol. 9, no. 5, p. 239, 2021, doi: 10.11648/j.ijsts.20210905.16.

[6] E. Weltevrede, A. Helmond, and C. Gerlitz, “The Politics of Real-time: A Device Perspective on Social Media Platforms and Search Engines,” *http://dx.doi.org/10.1177/0263276414537318*, vol. 31, no. 6, pp. 125–150, Jun. 2014, doi: 10.1177/0263276414537318.

[7] J. D. Cryer and K.-S. Chan, *Time Series Analysis: With Applications to R*. 2008. doi: 10.1007/978-0-387-75959-3/COVER.

[8] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. 2018. Accessed: Mar. 19, 2023. [Online]. Available: https://otexts.com/fpp2/

[9] C. A. Sims, “Macroeconomics and Reality,” *Econometrica*, vol. 48, no. 1, p. 1, Jan. 1980, doi: 10.2307/1912017.

[10] C. Hughes, “Understanding the Spotify Web API,” *Engineering at Spotify*, Mar. 09, 2015. https://engineering.atspotify.com/2015/03/understanding-spotify-web-api/ (accessed Mar. 19, 2023).

[11] E. B. Dagum and S. Bianconcini, *Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-31822-6.

[12] R. B. Cleveland, W. S. Cleveland, and I. Terpenning, “STL: A Seasonal-Trend Decomposition Procedure Based on Loess,” *J Off Stat*, vol. 6, no. 1, Mar. 1990.

Code Availability

As this study uses many computer programming, an extra reference is included to show the codes involved in the making of this paper. https://github.com/daffaadra/Portofolio/tree/Coding-Portofolio/Undergraduate%20Thesis%20(Skripsi)