EDUCATUM JSMT Vol. 12 SPECIAL ISSUE (2025)
ISSN 2289-7070 / e-ISSN 2462-2451(17-32)
https://ejournal.upsi.edu.my/index.php/EJSMT/index

K-Nearest Neighbor Regression for Predicting Song Popularity Using Gower Distance

Hazmira Yozza[1,2], Riswan Efendi[1*], Nor Azah Samat[1], Izzati Rahmi[1,2], and S.M. Aqil Burney[3]

[1]Department of Mathematics, Universiti Pendidikan Sultan Idris, Malaysia
[2]Department of Mathematics and Data Science, Universitas Andalas, Indonesia
[3]Department of Computer Science, University of Karachi, Pakistan

**Corresponding author: riswanefendi@fsmt.upsi.edu.my*

Received: 1 Jan 2025; **Revision:** 20 February 2025; **Accepted:** 25 March 2025; **Published:** 28 April 2025

To cite this article (APA): Yozza, H., Efendi, R., Samat, N. A., Rahmi, I.., & S.M., A. B. (2025). K-Nearest Neighbor Regression for Predicting Song Popularity Using Gower Distance. EDUCATUM Journal of Science, Mathematics and Technology, 12, 17-32. https://doi.org/10.37134/ejsmt.vol12.sp.3.2025

To link to this article: https://doi.org/10.37134/ejsmt.vol12.sp.3.2025

Abstract

The machine learning approach is widely used to investigate human activities, such as in the art field. In the music industry, a song's popularity is essential to predict before it is released. In this paper, we were interested in predicting the popularity of songs using the K-nearest neighbor regression. The Spotify app was used to gather some information related to the audio features of a song, i.e., song duration, instrumentalness, loudness, acousticness, danceability, energy, liveness, speechiness, audio valence, key, audio mode, tempo, and time signature. This research used mixed-type variables; thus, the dissimilarity is measured using the Gower distance. In addition, two weighting methods were also compared to predict song popularity. Using 10-fold cross-validation, we found that the inversely proportional weights-distance showed better prediction performance when compared with equal weight. Moreover, we also found the best performance in predicting the song popularity is obtained when $k = 5$ nearest neighbors were used, with mean square error (MSE) of 636.75 and mean absolute percentage error (MAPE) of 41.58% that implies a reasonable prediction result.

Keywords: song popularity, k-nearest neighbor regression, audio feature, Gower distance, weighting method

INTRODUCTION

In the current digital music industry, song popularity can be measured based on statistics such as the number of downloads, listeners, and play counts. Predicting the popularity of a song is essential and exciting before it is released since it provides valuable information for artists and record labels to promote marketing and promotion strategies, including estimating the time to release the new album, connecting singers' efforts to public interest, and so on [1]. For artists, predicting song popularity has an advantage in understanding song audio properties that drive popularity. In the next step, this can help artists improve their ability to produce a more commercial song. One way is by investigating factors that have a significant contribution to the success of songs. Predicting song popularity is also useful for music streaming service providers. Accurate song popularity prediction enables music providers to optimize their marketing strategies by tailoring playlists containing songs that are predicted to be popular.

Predictions of a song popularity can be approached by considering two primary sources of information, internally and externally. External factors are sourced from social and commercial aspects, such as album cover design and promotion. In contrast, internal factors are related to the song's content, including artists, song lyrics, genre, and audio properties [2,3]. While external factors are undoubtedly important, it is also essential to consider the internal aspects of a song when determining its popularity.

That is why studies exploring the relationship between a song popularity and its audio features are becoming a significant area of interest in music analytics. These studies are possible due to various music streaming service providers such as the Spotify Application Programming Interface (API) or a repository of song data such as the Million Song Dataset that provides the data needed.

Some research investigated how the audio characteristics of a song contribute to its popularity in the music market. Among the earliest is research conducted by [4] that evaluated the power of several classification methods, namely linear discriminant analysis (LDA), neural networks, support vector machine (SVM), and logistic regression in predicting the popularity of a song. Studies in this area are also conducted by [2] and [5] to predict the success of a song using songs listed in the Billboard music cart. Another research used a song popularity dataset scrapped from Spotify and compared several classification techniques, namely k-nearest neighbor classifier, linear SVM, and random forest classifier, to predict whether a song will be hit [6].

The majority of studies on the prediction of song popularity employ classification techniques. These techniques entail the conversion of the song popularity score from a value of 0-100 into two categories: popular and unpopular. One of the potential limitations of this approach is the loss of information related to the popularity score of a song due to the binary transformation performed. For this reason, some researchers conducted a study to analyze the effect of song audio features on popularity from the perspective of regression analysis, as opposed to classification [7,8,9].

The ordinary least square (OLS) regression is predominantly used, although other machine learning methods, such as the *k*-Nearest Neighbor (KNN) regression, random forest regression, support vector machine regression (SVR), and neural network regression, can also be used to analyze song popularity dataset. The KNN is a supervised machine learning approach since it infers a 'learner' from a dataset called training data to determine the most k-similar observations that will be used as the basis in predicting the output for a new observation at a later stage [10]. It is a popular machine learning method for regression and classification tasks because of its flexibility, computational efficiency, and interpretability. Moreover, it is a nonparametric approach. Instead of making assumptions in data modeling as in ordinary regression analysis, this method lets the data more directly drive predictions [11].

Random forest regression is a development of the CART method. Random forest is an ensemble of decision trees generated by randomly selecting data and variables in an iterative bagging bootstrap sampling. The response prediction for a given data is obtained by combining the prediction made from each tree [12,13]. Random forest can handle nonlinearity but is computationally intensive for a large dataset. Support vector machine regression works by determining a hyperplane such that the predicted response value has less than ε deviation from its actual value. This method is effective for highly dimensional data [14]. The neural network is a massively parallel distributed processor comprising simple processing units known as neurons that naturally possess a proclivity for storing knowledge and ensuring its availability for utilization. It is designed in a similar way to how the brain executes a certain task or function of interest. It mimics the brain in two ways: the network acquires the knowledges from the environment through a learning process, and the strength of interneuron connection is employed to store the gained knowledge. It is flexible in modeling a complex relationship, can handle nonlinearity, and is adaptive to environmental changes. However, it needs a large amount of data to perform well since it employs a massive interconnection of neutrons[15].

Compared to neural network regression and support vector machine, the KNN and random forest regression are easy to implement and efficiently handle high dimensional data [12]. The KNN regression provides a better prediction than other regression methods in several applications. Research conducted by [16], which compared several regression approaches to predict suspended sendiment concentration in a river, found that the KNN regression provides a more reliable prediction. Other studies also found that the KNN regression outperforms the support vector regression and random forest [17].

The KNN regression is a development of the KNN classifier, which was initially implemented by Cover and Hart in 1967; thus, it uses steps similar to those used in the KNN classifier. A formal definition

of the KNN regression is presented below. Let $X = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N_1}$ is a training dataset that is formed by N_1 samples, each sample $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{im})$ is a vector of independent variables that contains m features, and y denotes the response variable. If a new query sample z is given, the predicted response variable for that new sample is determined by averaging the response value of its k nearest neighbors.

The parameter k is a substantial parameter in the KNN regression. It describes the number of nearest neighbors used to predict the new data's response. This parameter affects prediction accuracy and, thus, should be selected carefully. In determining the optimal value of this parameter, some studies suggested running the KNN algorithm repeatedly with different k and choosing the one that provides the highest accuracy [16,18].

The performance of the KNN algorithm also depends on the similarity or dissimilarity measure employed to describe the distance between objects and to define the meaning of 'near.' Consequently, choosing the suitable similarity/dissimilarity measure is crucial. Basically, the KNN algorithm is flexible in determining the similarity or dissimilarity measure. Several choices are possible but should be selected ad hoc, depending on the application. The Euclidean distance is a commonly used dissimilarity measure [19]. However, this distance measure is only suitable for numeric variables. The Euclidean distance is inappropriate for categorical and mixed-type data (a mixture of categorical and numeric variables) since its computation relies on numerical values. Categorical data does not possess an absolute order. The value assigned to a category is merely a code that represents the category. This fact makes the algebraic computations performed on categorical data meaningless. For this reason, when the Euclidean distance and other dissimilarity metrics that rely on numerical values are applied to categorical or mixed data, it can lead to a misleading result [20].

The Gower dissimilarity coefficient or the Gower distance is a more suitable dissimilarity measure for mixed-type data[21,22]. It provides a simple and elegant way to measure similarity between two instances [23] and performs well for inconsistent information [24]. The Gower distance can handle missing values. This distance measure can still be calculated despite missing value on one or more variables [22]. Although other dissimilarity measures are available for mixed data, the Gower distance is more popular. It is incorporated in the Daisy function in the R package for clustering and applied in the clustering task in [25]. The Gower dissimilarity coefficient in the KNN algorithm was employed for the classification in [24,26].

This research aimed to determine KNN regression rules to predict song popularity in Spotify streaming based on some audio features. The rules include the optimal number of nearest neighbors. The novelty of this research is related to the dissimilarity measure employed in the KNN regression. The KNN regression was performed to predict song popularity by Dong et al. [9]. However, in that study, as in many other studies, the distance measure employed was the Euclidean distance, despite the data being mixed type. In this research, we will slightly modify that research by using a more appropriate dissimilarity measure, namely the Gower distance. In addition, this research was also intended to compare two weighting methods used in averaging the nearest neighbors' responses to predict the popularity score of a given song and determine the best weighting method.

DATA AND METHOD

Data Collection

The dataset used to find the rule for predicting the song popularity is the song popularity dataset scrapped by M.Yaser H from Spotify API and shared as a public dataset on www.kaggle.com [27]. The original data contains 18,835 song tracks. Data was cleansed by removing duplicate and invalid data, resulting a dataset comprising 14.360 song tracks. Furthermore, due to computational limitations, this research only uses 5,000 song tracks randomly selected from the cleansed dataset. Referring to Slovin's formula, this sample dataset is considered to represent of the entire data set with a considerably small margin of error

(<0.0125).

The response variable is song popularity, which is defined as the track's popularity and score from 0 to 100, and a score of 100 is the most popular. We use 13 songs' audio features as explanatory variables, which is mixed-type data. The definition and type of variables are listed in Table 1.

Table 1 List of dependent variables

Variable	Definition [8, 28]	Data type
Danceability	It describes how suitable a track is for dancing based on a combination of musical elements, including tempo, rhythm stability, beat strength, and overall regularity. Values range from 0.0 to 1.0, with 0.0 for the least danceable and 1.0 for the most danceable.	Interval
Audio valence	It describes the musical positiveness conveyed by a track. The score varies from 0 to 1.0, where the more positive track (e.g., sounds happy, euphoric, and cheerful) will have a high valence score. On the other hand, a track that sounds more negative (e.g., angry, depressed, sad) will have a low value of valence.	Ratio
Energy	It describes a perceptual measure of intensity and activity. The score is in an interval range of 0.00-1.00. Generally, energetic tracks feel fast, loud, and noisy. Energetic tracks like Death Metal will have a high energy score. On the other hand, a non-energetic track, such as a Bach prelude, will have a low score on the scale. This score is calculated from several features, e.g., perceived loudness, onset rate, dynamic range, timbre, and general entropy.	Ratio
Tempo	It measures the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the pace or speed of a given piece. Tempo derives directly from the average beat duration.	Ratio
Loudness	It measures a track's overall loudness in decibels (dB). Loudness score are averaged across the entire track and are valid for comparing the relative loudness of tracks.	Ratio
Speechiness	It detects whether spoken words are present in a song track. The more exclusively speech-like recording (talk show, poetry, audiobook, etc.), the closer speechiness score to 1.0.	Ratio
Instrumentalness	It predicts whether a track contains no vocals. In this context, non-verbal sounds (e.g., ooh or aah) are instrumental; Conservaly, rap, or spoken word tracks are clearly vocal**.**	Ratio
Liveness	It detects the presence of an audience in the recording. Higher liveness values represent an increased possibility that the track was recorded live	Interval
Accousticness	It measures the confidence measure of whether the track relies more on acoustic instruments or electronics. It measures from 0.0 to 1.0	Interval
Key	It describes the estimated overall key of the track. Integers map to a nominal pitch using standard Pitch Class integer notation, namely 0 = C, 1 = C♯/D♭, 2 = D, 3= D♯/E♭, 4=E, 5=F, 6= F♯/G♭, 7=G, 8= G♯/A♭, 9=A, 10= A♯/B♭, and 11=B.	Nominal
Audio mode	It indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1, and the minor is 0.	Nominal
Song duration	It measures the duration of the track (in milliseconds)	Ratio
Time signature	It is used as a track's estimated overall time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).	Nominal

In Table 1, all involved variables are defined and explained, including mode, key, and time signature and their types. Interestingly, the relationship between response and independent variables is also

illustrated using a variable framework in Figure 1.

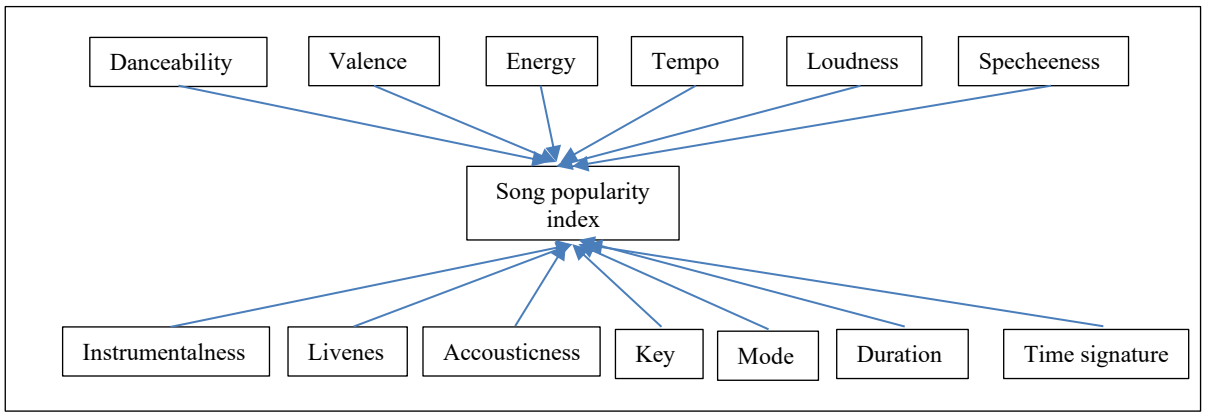


Figure 1. Variable framework for song popularity

Phase of Data Analysis

Data analysis is divided into three different phases as follows:

Phase 1: Split dataset into training and testing datasets. Let $\mathbf{X} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N_1}$ be a training dataset comprised of N_1 samples with $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{im})$ is a m-dimension vector of independent variables. Let $\boldsymbol{z}$ be an m-dimension vector of independent variables for a new sample in the testing dataset. Apply the KNN regression to predict the response of new observation $\boldsymbol{z}$ using the following steps.

1. Set k ($1 \leq k \leq N$), the number of nearest neighbors used in prediction;
2. Calculate distance between every $\boldsymbol{z}$ in the testing dataset and $\boldsymbol{x_i}$ for all i = 1, 2, …, N. Since the dataset is a mixed-type, this research implemented the Gower distance to measure the distance between pairs of samples. The Gower distance can be determined as follows. Denote x_c as the c-th variable and d_{izc} is a dissimilarity measure between $\boldsymbol{z}$ and $\boldsymbol{x_i}$ for the c-th variable (c = 1, ..., m).
   - If x_c is nominal, the dissimilarity between two samples is expressed as

$$d_{izc} = \begin{cases} 1, & x_{ic} \neq x_{zc} \\ 0, & x_{ic} = x_{zc} \end{cases} \quad (1)$$

   - If x_c is numerical (ratio/interval), the dissimilarity between two samples is

$$d_{izc} = \frac{|x_{ic} - x_{zc}|}{\max(x_c) - \min(x_c)} \quad (2)$$

     where $\max(x_c)$ is the maximum value of x_c and $\min(x_c)$ is its minimum value.
   - If x_c is ordinal, transform all categories using the formula

$$x_{ic} = \frac{r_{ic} - 1}{R_c - 1} \quad (3)$$

     where r_{ic} is the rank number of the i-th ordinal category (r = 1, ..., R_c) and R_c the maximum rank number of x_c. The dissimilarity between two samples for the c-th variable is calculated based on these transformed values using the formula for numeric variables.

The Gower distance between every $\boldsymbol{z}$ and $\boldsymbol{x_i}$ is calculated from the equation

$$d_G(\boldsymbol{z}, \boldsymbol{x_i}) = \frac{\sum_{c=1}^{m}(w_{izc} d_{izc})}{\sum_{c=1}^{m} w_{izc}} \quad (4)$$

where w_{ijc} = 0, if the value of c-th variable for either $\boldsymbol{z}$ *or* $\boldsymbol{x_i}$ is missing; otherwise, w_{ijc} = 1 [29].

The computation of Gower distances can be illustrated as follows. Suppose a dataset consists of n

observations that are measured by four variables: X_1, X_2, and X_3 that are measured on a categorical nominal, categorical ordinal with categories 1-4, and interval/ratio scale, respectively. Suppose that the minimum and the maximum values of X_2 are 1 and 6. Table 2 shows the values of three observations in the dataset.

Table 2. Data example

Observation	X_1 nominal	X_2 Interval/ratio	X_3 ordinal
A	2	3.2	3
B	1	4.7	4
C	2	*	2

Dissimilarity of A and B based on X_1 and X_2 computed using eq(1) and eq(2) are:

$$d_{AB1} = 1.$$

$$d_{AB2} = \frac{|x_{A2} - x_{B2}|}{\max(x_2) - \min(x_2)} = \frac{|3.2 - 4.7|}{6.0 - 1.0} = 0.3$$

To compute the dissimilarity between A and B based on X_3, all categories of X_3 need to be transformed. Given that the maximum rank is 4, eq(3) transforms categories 1,2,,3, and 4 into values 0, 1/3, 2/3, and 1, respectively. Thus, the dissimilarity between A and B for X_3 is

$$d_{AB3} = \frac{|x_{A3} - x_{B3}|}{\max(x_3) - \min(x_3)} = \frac{|2/3 - 1|}{1 - 0} = 0.33.$$

Hence, the Gower distance between A and B is

$$d_G(\boldsymbol{A}, \boldsymbol{B}) = \frac{\sum_{c=1}^{3}(w_{ABc} d_{ABc})}{\sum_{c=1}^{3} w_{ABc}} = 0.54.$$

Here, $w_{ABc} = 1$ for all c since all values are non-missing. Following the same procedure, dissimilarity between A and C for X_1 and X_3 are $d_{AC1} = 0$ and $d_{AC3} = 0.33$, while the dissimilarity based on X_2 are not computed due to its missing value. Therefore, in calculating the Gower distance between A and C, $w_2 = 0$ and the Gower distance is $d_G(\boldsymbol{A}, \boldsymbol{C}) = 0.165$.

3. Sort distances from $\boldsymbol{z}$ to every $\boldsymbol{x_i}$ in the training dataset, in ascending order; find a subset of **X** that contains k nearest neighbors of $\boldsymbol{z}$. The term 'nearest' means that $\boldsymbol{x_i}$ has the smallest distance to $\boldsymbol{z}$.
4. Predict the value of the response variable of new observation, $\hat{y}(\boldsymbol{z})$, by determining a weighted average of dependent variable values of its nearest neighbors, formulated by:

$$\hat{y}(\boldsymbol{z}) = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i} \quad (5)$$

where w_i represents the weight assigned to i-th nearest neighbors when used to estimate the response variable of a particular observation; y_i denotes the response value of i-th nearest neighbors, and k denotes the number of nearest neighbors.

This research compared two weighting factors, that is

- $w_i = 1$; in this case, the predicted value of the dependent variable for a new observation is calculated by simply averaging the dependent variables of its k nearest neighbors.
- The weight assigned to the i-th observation is calculated by

$$w_i = \frac{d_{i(rel)}}{\sum_{i=1}^{k} d_{i(rel)}} \quad (6)$$

where:

$$d_{i(rel)} = \left(\frac{\sum_{i=1}^{k} d_i}{d_i}\right)^p, \tag{7}$$

d_i is the distance between $\boldsymbol{z}$ and $\boldsymbol{x_i}$, and p is a power parameter that considers different forms of distance–weight relationships [16,30]. This research is a preliminary study to compare the KNN regression with equal and unequal weights. Therefore, this research employed only one possible value of p, namely p =1, which is considered the easiest and the simplest p to calculate the weight.

Phase 2: Calculate MAPE and MSE as the measure of prediction accuracy. The MAPE is the mean absolute percentage error and is computed using the formula of

$$MAPE = \frac{1}{N}\left(\frac{|\hat{y}_i - y_i|}{y_i}\right)\%, \tag{8}$$

while MSE is the mean of square error and is calculated using

$$MSE = \frac{1}{N}(\hat{y}_i - y_i)^2 \tag{9}$$

Phase 3: Selecting the optimal k-value

The number of nearest neighbors (k) is vital in the KNN regression. To select the optimal k, a 10-fold cross-validation is implemented using the following steps. This method also ensures that the prediction is not underfitting or overfitting.

a. Randomly split data into ten sub-datasets;
b. Iteratively, repeat Phase 1 using a sub-dataset as a testing dataset and the remaining data as a training dataset. From this step, we obtain the prediction of responses for all observations in the dataset;
c. Calculate the MAPE and MSE of all observations in the dataset;
d. Repeat steps a – c for $k = 1, 2, \ldots, 155$, then select k that gives the smallest MAPE and MSE as the optimal k.

Results obtained from two cases of the KNN regression, with equal and unequal weights, will be compared to determine the best weight used to predict song popularity. The comparison was conducted using line graphs illustrating the MSE or MAPE for k=1, 2, …, 155. In addition, paired t-tests were also performed to test whether the mean of MSE and MAPE values obtained in the KNN regression with unequal weight were lower than those obtained in the KNN regression with equal weight. These results will also be compared to predictions carried out by ordinary linear regression.

RESULTS AND DISCUSSION

Data Pre-processing

The data gathered from any source are often inconsistent, noisy, or even incomplete, especially when the data are collected data from several clients or combined from a varied source or scrape data. Also, there is the possibility of creating duplicate data. This condition will lead to data analysis problems and weaken the model's predictive capability. Therefore, cleansing the data should be the initial stage in every machine-learning task. The data cleansing procedure involves identifying noisy, incomplete, and incorrect data in the dataset. In the following stage, these data are corrected or removed from the dataset. There are several ways: removing irrelevant or duplicate data, fixing the structural error, and handling the missing value problem [31].

In this research, the cleansing process includes removing all duplicate data. In addition, data

cleansing is also done by removing observations with invalid values in one or more features. For instance, the time signature variable is the number of beats per bar, and the valid values for time signature are 3-7. Therefore, observations with time signature values of 0 or 1 are considered invalid. Since there was no further information to correct this error, observations with invalid values were removed from the dataset. We also removed observations with a popularity score of zero. After the cleaning process, 14.396 tracks are ready for further analysis.

Explanatory Data Analysis

Explanatory data analysis is performed to get insight into the distribution of variables involved. The histogram and density plot of popularity score for the whole dataset is described in Figure 2.

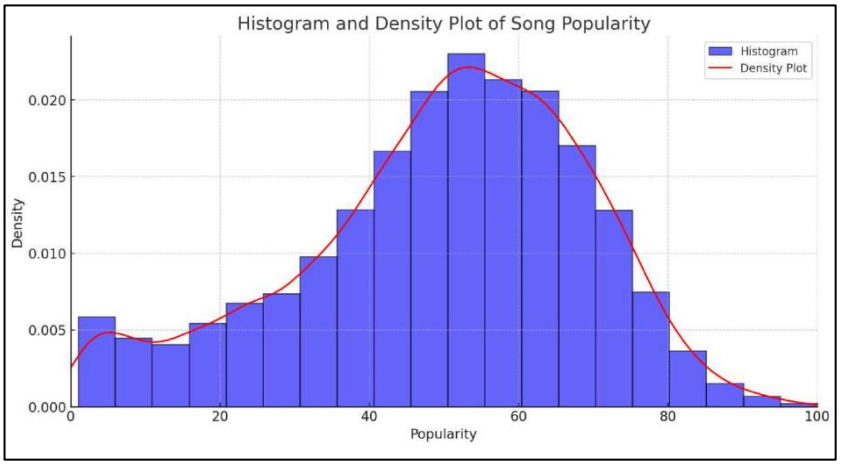


Figure 2. The histogram and distribution of song popularity

Figure 2 illustrates the distribution of song popularity values within the dataset. It shows that the majority of songs exhibit relatively low to medium popularity. There are a few songs that have achieved a high level of popularity. However, they represent a relatively small proportion of the songs. This pattern is commonly observed in song popularity datasets, where only a small percentage of songs attain a high popularity score.

Figure 3 illustrates the distribution of numerical variables used as explanatory variables. From this figure, it is observed that the majority of songs have a zero score in acoustics. This variable measures how acoustic the track is, so it is clear that most songs in the dataset rely solely on electronic instruments. Danceability represents the suitability of a song track for dancing. The higher the score, the more danceable the track is. Figure 3 shows that most songs have danceability scores between 0.6 – 0.8. This result indicates that most songs are enjoyable for dancing since they have a consistent and strong beat that makes them more energetic. The histogram for energy gave a similar conclusion where most songs have a high energy score, reflecting that those songs are energetic.

The instrumentalness measures how likely a track is instrumental with no vocal presence; the higher the score, the greater the probability that the track is instrumental, and vice versa. In Figure 3, it is observed that around 90% of tracks have zero scores in instrumentalness. This result indicates that almost all songs are not instrumental and contain vocals like singing. Liveness detects the presence of people, applause, and other noise while recording the song. In other words, this feature predicts whether a song is a live performance or studio-made. Liveliness above 0.8 reflects a high possibility the song was recorded live [32]. The liveness histogram shows that most songs have low to moderate liveliness scores.

It reflects that most songs were recorded in non-live settings with minimal audience presence. A few songs have high liveliness scores, which suggests that they are recorded in a lively setting with a more dynamic performance with a larger audience.

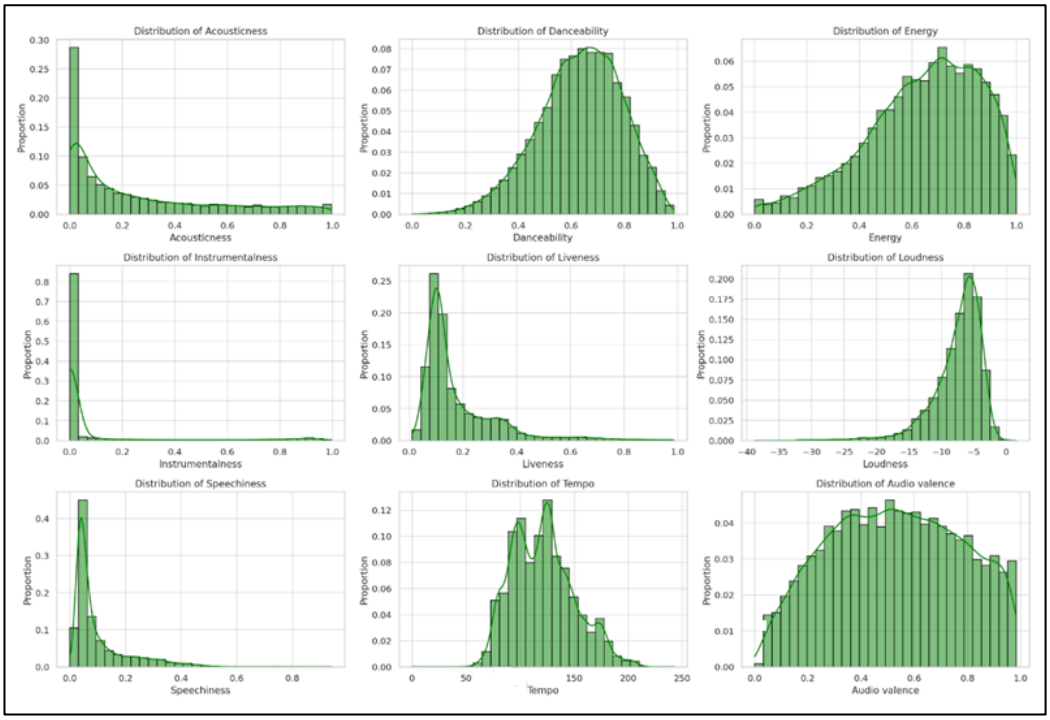


Figure 3. Distribution of numerical song's audio features

The loudness of a track measures the overall intensity of the sound. It is computed by averaging the sound pressure levels across the entire track and has a value from -60 decibels (dB) to 0 dB. The loudness helps compare the relative loudness of the tracks. Based on the histogram of loudness, it is known that the distribution of tracks' loudness is skews negative. The majority of songs have a value between -10 dB and -6 dB, indicating that most songs are loud. A few songs are quieter, softer, or more mellow if compared to other songs with loudness less than -15. The speechiness detects whether spoken words are present in tracks.

The speechiness > 0.66 implies that the track contains entirely spoken words; conversely, a track with speechiness < 0.33 is probable music only without significant spoken words. Tracks with speechiness score in the interval of 0.33-0.66 may contain music and spoken words, e.g., hip-hop or rap. The histogram of speechiness shown in Figure 3 indicates that most songs were entirely made by music, with speechiness of less than 0.33. A small part of songs contains music or other speech-like sounds, as is usually found in hip-hop or rap music.

The tempo of a song is the speed of the song, indicating how quickly the song when it is played. High-tempo songs generally have higher energy when compared to a song with a lower tempo [33]. The tempo of songs in this dataset is observed to fall between 50 to 240 BPM (beats per minute) but concentrates in a moderate range, between 80 to 120 BPM.

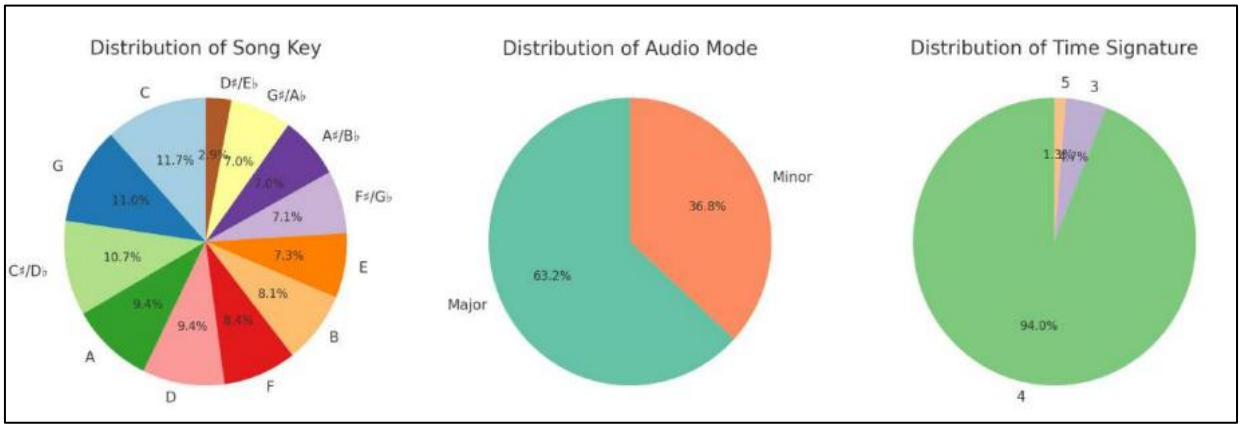


Figure 4. Pie chart of categorical song's audio features

In Figure 4, The distribution of musical keys is spread across all 12 possible keys. No single key significantly dominates the dataset, suggesting a diverse range of musical compositions in pitch and tonality. Nevertheless, the most frequently used among the twelve keys correspond with 0, 1, and 7, i.e., keys C, C♯/D♭, and G, respectively. This figure also shows that the mode is predominantly in the major audio mode (mode=1). Around 63% of the songs are in this mode. The major audio mode is often associated with a happy and energetic song, while a song with minor audio modes conveys a sad feeling. This result indicates that more upbeat and energetic songs are often used than sad songs. Regarding time signature, this figure demonstrates that most songs (approximately 96%) have a 4/4 time signature, the most common time in music.

Multiple Regression Model

The relationship between popularity scores and related independent variables is analyzed using multiple linear regression models. Furthermore, the parameters of song popularity were estimated by ordinary least squares (OLS). Mathematically, the regression equation is written in Eq. (10).

$$Score = 59.72 + 0.000001\,\mathrm{SD} - 3.56\,\mathrm{Ac} + 5.69\,\mathrm{Da} - 9.55\,\mathrm{En} - 6.24\,\mathrm{In} - 4.19\,\mathrm{Li} + 0.45\,\mathrm{Lo} - 2.02\,\mathrm{Sp} - 0.012\,\mathrm{Te} - 5.56\,\mathrm{AV} + 1.66\,\mathrm{Key}_1 \ldots + 1.72\,\mathrm{Key}_{11} + 0.80\,\mathrm{AM} + 1.68\,\mathrm{TS}_4 + 3.36\,\mathrm{TS}_5 \quad (10)$$

where SD is song duration; Ac is acousticness; Da is Danceability; En is Energy; In is instrumentalness; Li is liveness; Lo is loudness; SP is speechiness; Te is tempo, AV is audio valence; AM is audio mode with minor mode as the reference; Key_j $(j = 1, 2, \ldots, 11)$ are dummy variables for Key features, with Key_0 as the reference. TS_j $(j = 4{,}5)$ are dummy variables for Time signatures with $\mathrm{Time\_signature}_3$ as the reference.

As shown in Eq. (10), liveness negatively impacts a song's popularity score. This result implies that a song recorded in a live performance setting or containing more sound like in a live performance tends to have a low popularity score. Consequently, such kinds of songs will be less popular. A negative coefficient for instrumentalness means that a song with fewer or no vocals and more instrumental content tends to be less popular. In addition, the negative value of the acousticness parameter represents that songs that depend more on acoustic instruments also have a lower popularity score and, hence, are less popular. On the contrary, danceability positively affects the popularity score. It means that the higher the danceability score, the more suitable that song is for dancing, and the higher the possibility that the song become popular.

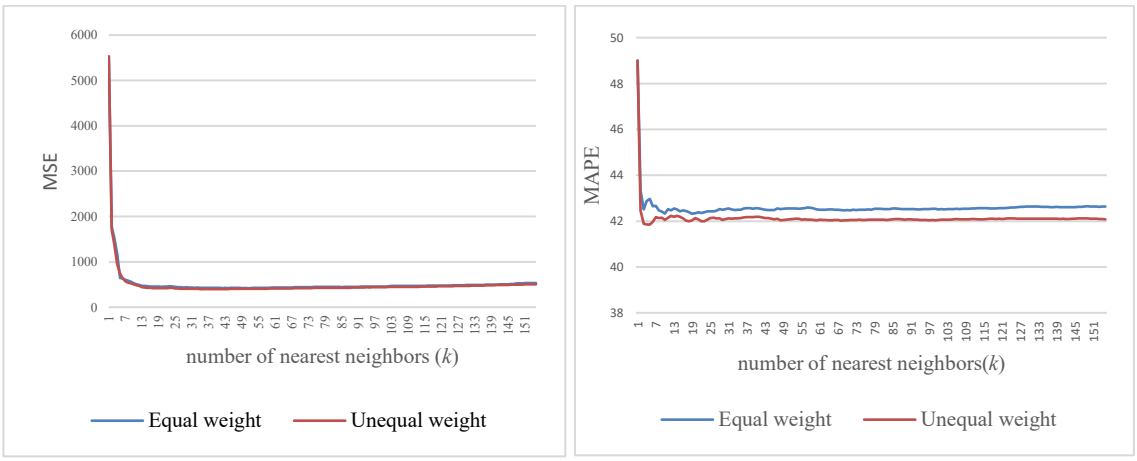
The t-test performed to test the significance of audio features' influence on the song popularity score shows that all audio features except song duration and speechiness affect the score significantly at a level

of significance $\alpha = 10\%$. However, this model can only explain 22.47% of the song popularity variation. Model assumptions checking shows that the resulting model violates normallity and homoscedasticity assumptions.

K-Nearest Neighbors Regression Performance

This research tried to determine rules in the KNN regression to predict the popularity of a song streamed on Spotify based on some audio features. This dataset uses mixed data. Therefore, instead of using the Euclidean distance measure commonly used in many k-nearest neighbor applications, this study utilized the Gower distance to measure the distance between observations. The corresponding rules investigated include the number of nearest neighbors used in prediction and whether utilizing equal or unequal weights is better.

The prediction of the popularity score of a new given song using the KNN involves averaging the popularity score of k songs, referred to as the 'nearest neighbors', using either a weighted or unweighted average. Actually, in the case of an unweighted average, an equal weight is assigned to every nearest neighbor; thus, the prediction of the popularity score of a particular song is calculated by simply averaging the popularity scores of its k nearest neighbors. In the case of a weighted average, each nearest neighbor is given unequal weight. The most appropriate weight to be assigned is the distance between a given song and its nearest neighbor. The weight attributed to a neighbor is inversely proportional to the distance between them, with greater distances resulting in smaller weights. The following figure compares the performance of k-nearest neighbor regression when using equal and unequal weights for up to $k = 155$ nearest neighbors. The comparison criteria used are the mean squared error (MSE) and the mean absolute percentage error (MAPE) values of the prediction results obtained through a 10-fold cross-validation method.



a. Mean square error (MSE) b. Mean absolute percentage error (MAPE)

Figure 5. Comparison between prediction performance of KNN regression with equal and unequal weight for various k based on (a) MSE and (b) MAPE

Figure 5 compares the performance of KNN regression with equal and unequal weight in predicting song popularity for various numbers of nearest neighbors. The blue line represents the prediction performance of the KNN regression with equal weight for k =1, 2, …,155, while the red line represents the performance of the KNN regression with unequal weight. The comparison is conducted based on two criteria: mean square error (Figure. 5a) and mean absolute percentage error (Figure 5b). Figure 5a illustrates that the mean square error (MSE) value generated by the K-nearest neighbor regression with equal weights is marginally higher than that produced by the KNN regression with unequal weights. This

discrepancy in performance is more pronounced when examining MAPE values for both scenarios. Using unequal weights also yields a lower MAPE value for predicting song popularity.

In addition, a paired t-test was performed to test whether the mean of MSE obtained in the KNN regression with unequal weight is lower than that obtained in the KNN regression with equal weight, or

$$H_0 : \mu_{unequal} = \mu_{equal}$$
$$H_1 : \mu_{unequal} < \mu_{equal}$$

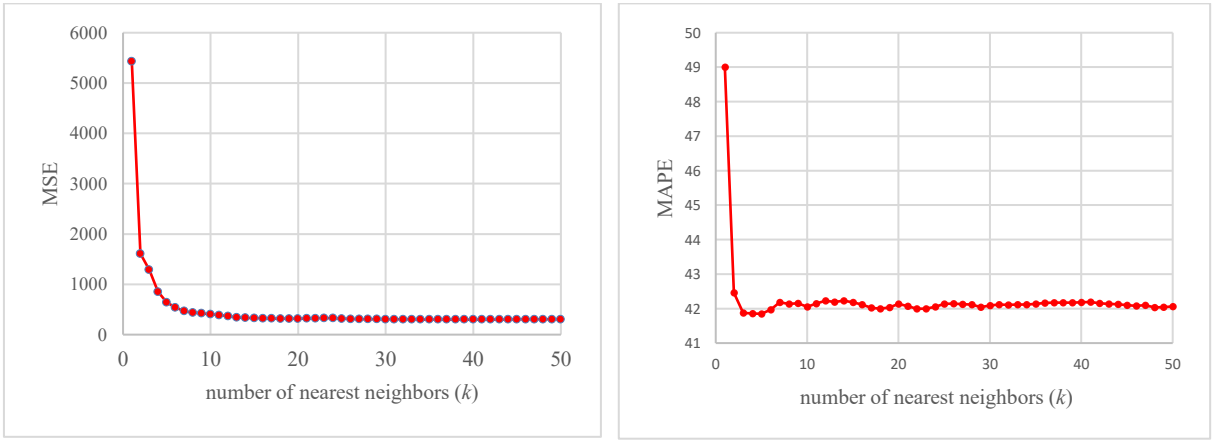
The same test was also carried out for MAPE. Results are presented below in Table 3.

Table 3. Result of t-test

Criteria	t-value	p-value
MSE	-5.85	0.000
MAPE	-15.70	0.000

The significance level (α) used is 5%. Table 3 shows that the t-test for MSE yielded p-values lower the significance level. Thus, the null hypothesis is rejected, and it can be concluded that the mean of MSE values obtained in the KNN regression with unequal weight is lower than that obtained in the KNN regression with equal weight. Hypothesis testing of MAPE value gave a similar conclusion. These results validate the previous conclusion regarding implementing equal and unequal weight in the KNN regression. Therefore, it is concluded that employing unequal weights in KNN regression is more favorable than equal weights in estimating a song's popularity.

The number of nearest neighbors (k) employed in predicting the response is another parameter in k-nearest neighbor regression. Thus, in the following stage, the optimal number of k^* will be determined. In this study, k^* is identified by performing the k-nearest neighbor regression for various values of k ($k = 1, \ldots, 50$). For each case, estimation performance measures are calculated and evaluated. The value of k that yields the most favorable results is then selected as k^*. As with determining weights used, the performance measures employed are MSE and MAPE, which are determined through a 10-fold cross-validation procedure. The k^* is the k value that produces the lowest MSE and MAPE. The performance of KNN regression using inversely proportional weight for $k = 1, 2, \ldots 50$ nearest neighbors are depicted in Figure 6.



b. Mean square error (MSE)

b. Mean absolute percentage error (MAPE)

Figure 6. Prediction performance of KNN regression for various k based on (a) MSE and (b) MAPE

Figure 6 illustrates the performance of the KNN regression in predicting popularity score based on MSE and MAPE, respectively. Figure 6a shows that KNN regression prediction using $k = 1$ nearest

neighbor gives a high MSE. The MSE value declines rapidly until $k = 5$ and continues to decrease for the greater k. However there is no substantial reduction in MSE. Thus, based on the MSE, $k = 5$ is chosen as k optimal. The right-hand line chart (Figure 6b) depicts the prediction performance based on MAPE. As in the previous chart, for $k = 1$, the KNN regression yields a high MAPE. This value declines for greater k and reaches a minimum value for $k = 5$. Thus, based on MAPE, $k = 5$ is also chosen as an optimal k.

As a result, based on those predicting performance measures, the optimal number of nearest neighbors used as the basis of prediction in the KNN regression is $k^* = 5$. The best result is obtained using an unequal weight inversely proportional to the Gower distance from a given song to each nearest neighbor. With these scenarios, the KNN regression provides an MSE of 636 and MAPE of 41.584%. This MAPE value implies a reasonable prediction result. Compared to the least square regression, which explains a relatively small amount of song popularity variation, the KNN regression is considered to provide a better prediction.

Despite the prediction performance exhibiting a MAPE of 41.58%, this analysis still provides valuable insight, primarily when associated with a highly diverse range of song listener's characteristics with diverse preferences. The conclusion related to the weighting method applied and the number of nearest neighbors used can be referred to in the further KNN regression application that is performed to predict the popularity score of a given song. By implementing these rules, the streaming platform can predict the popularity score of a new song and create a new playlist consisting of songs likely to achieve commercial success and other songs with similar characteristics. On the other hand, artists and music labels can apply these rules to identify songs with audio features identical to their new song, predict its success, and plan promotional strategies to enhance the popularity of their song.

This research finding, especially related to predictive performance level, allows a space for improvement. It is important to note that certain limitations were encountered during the research. A possible explanation for moderate-level performance is that this current research focuses on song audio features in predicting song popularity while ignoring other variables that probably have a more substantial influence on song popularity than those used in this study, such as lyrics, song genre, and other variables attached to the singer. Another possibility is the presence of uncertainty or fuzziness in the relationship between popularity score and song audio features. Integrating other statistical methods that can capture the fuzziness of the relationship between variables is imperative to be explored further.

Moreover, there are several possible sources of dataset bias. First, it is related to cultural bias. This type of bias arises as all songs in the dataset used in modeling focus on Western songs. Since song preference varies between listeners from different countries, cultures, and segments (age, etc.), this dataset fails to generalize across listeners' demographics, thus limiting their effectiveness in predicting popularity for non-western songs. Another bias is associated with the definition of 'popularity' itself. Popularity in this dataset is measured mainly based on the number of streaming. This definition may neglect songs that are popularized through word-of-mouth rather than the streaming platform. Further bias is popularity bias, which occurs when the recommendation system and algorithm prioritize popular songs, leaving less popular songs underrepresented in recommendation [34].

CONCLUSION

In this paper, KNN regression has been implemented to predict the popularity score of a song based on the song's audio features. The best scenario includes the number of nearest neighbors used as the basis

of prediction and the weighting method used in averaging the popularity scores of the song's nearest neighbor. The most favorable scenario with the lowest MSE and MAPE were found to be achieved using (a) unequal weight, which is inversely proportional to the distance from a particular song to its selected nearest neighbors, and (b) equal weight with $k = 5$ nearest neighbors. This scenario provides a MAPE of 41.5%, implying a reasonable predictive performance. With this level of accuracy, this current analysis can serve as a helpful tool to predict the popularity score of a given song. However, it also provides room for improvement by addressing limitations encountered in this research.

In line with the limitations explained before, some actions need to be taken in future studies to address those limitations, increase the predictive performance level, and provide a more comprehensive explanation regarding the relationship between song popularity and its explanatory variables. First, by using unequal weight, better performance was achieved if compared with equal weight. Using unequal weight involves another parameter, p, which is a power parameter. This parameter also needs to be considered in order to investigate different relationship forms between weight and distance. In this case, p was considered equal to 1. So, in the future study, the optimal p that performs better in predicting song popularity will be determined.

The second action related to variables used in the analysis. In this research, thirteen audio features were used to predict the popularity score of songs in this study. However, these features were insufficient to explain the popularity of songs precisely. In future research, unrelated features will be removed, while other additional explanatory variables that are likely to affect popularity scores better will be integrated. Those variables include song genre, lyric and voice (male or female voice). Incorporating these variables may enhance the model's accuracy and facilitate a more comprehensive understanding of the factors influencing song popularity. Another action is associated with the dataset used. For future research, we will focus on investigating the popularity of Indonesian songs and their related features. Finally, the implementation of the fuzzy linear regression will be explored to address the possibility of the presence of fuzziness in the relationship between popularity score and its explanatory variables. Another challenge is the possibility of constructing an alternative unbiased metric to measure the popularity of a song.

REFERENCES

[1] Araujo, C.V.S., Cristo, M.A.P., & Giusti, R. (2019). Predicting music popularity using music charts. *Proceeding of 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 859-864. https://doi: 10.1109/ICMLA.2019.00149.

[2] Yang, L.-C., Chou, S.-Y., Liu, J.-Y., Yang, Y.-H., & Chen, Y.-A. (2017). Revisiting the problem of audio-based hit song prediction using convolutional neural networks. *Proceeding of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 621-625. https://doi.org/10.48550/arXiv.1704.01280

[3] Al-Beitawi, Z., Salehan, M., & Zhang, S. (2020). What makes a song trend? Cluster analysis of musical attributes for Spotify top trending songs. *Journal of Marketing Development and Competitiveness*, *14*(3), 79-91. https://doi.org/10.33423/jmdc.v14i3.3065

[4] Pham, J., Kyauk, E., & Park, E. (2016). *Predicting song popularity* (Tech. Rep. Vol. 26). Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA. https://cs229.stanford.edu/proj2015/140_report.pdf

[5] Askin, N., & Mauskapf, M. (2017). What makes popular culture popular? Product features and optimal differentiation in music. *American Sociological Review, 82*(5), 910–944. https://doi.org/10.1177/ 0003122417728662

[6] Pareek, P., Shankar, P., Pathak, P., & Sakariya, N. (2022). Predicting music popularity using machine learning algorithm and music metrics available in spotify. *Journal of Development Economics and Management Research Studies (JDMS)*, 9(11), 10 -19. http://doi.org/10.53422/JDMS.2022.91102

[7] Suh, B. J. (2019). International music preferences: an analysis of the determinants of song popularity on Spotify for the US, Norway, Taiwan, Ecuador, and Costa Rica. *CMC Senior Theses.* https://scholarship.claremont.edu/cmc_theses/2271.

[8] Saragih, H.S. (2023). Predicting song popularity based on Spotify's audio features: insights from the Indonesian streaming users. *Journal of Management Analytics*, *10*(4), 693-709. https://doi.org/10.1080/23270012.2023.2239824

[9] Dong, A., Qiu, R., & Ye, Z. (2023). Regression analysis of song popularity based on ridge, K-nearest neighbors and multiple-layers neural networks. *Highlights in Science, Engineering and Technology*, *39*, 609-617. https://doi.org/10.54097/hset.v39i.6602

[10] Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, *251*, 26–34. https://doi.org/10.1016/j.neucom.2017.04.018

[11] Chen, G.H. & Shah, D. (2018). Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends in Machine Learning*, *10*(5-6), 337–588. https://doi.org/10.1561/2200000064.
Cosenza, D. N., Korhonen, L., Maltamo, M., Packalen, P., Strunk, J. L., Næsset, E., ... & Tomé, M. (2021). Comparison of linear regression, k-nearest neighbour and random forest methods in airborne laser-scanning-based prediction of growing stock. *Forestry: An International Journal of Forest Research*, *94*(2), 311-323. https://doi.org/10.1093/forestry/cpaa034

[12] Shataee, S., Kalbi, S., Fallah, A., & Pelz, D. (2012). Forest attribute imputation using machine-learning methods and ASTER data: comparison of *k*-NN, SVR and random forest regression algorithms. *International Journal of Remote Sensing*, *33*(19), 6254–6280. https://doi.org/10.1080/01431161.2012.682661

[13] Zhang, F., & O'Donnell, L. J. (2019). Support vector regression. *Machine Learning*, 123-140. https://doi.org/10.1016/B978-0-12-815739-8.00007-9

[14] Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Pearson Education, Inc., McMaster University, Hamilton. http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf

[15] Fathabadi, A., Seyedian, S.M., & Malekian, A. (2022). Comparison of bayesian, k-nearest neighbor and gaussian process regression methods for quantifying uncertainty of suspended sediment concentration prediction. *Science of The Total Environment*, *818*, article151760. https://doi.org/10.1016/j.scitotenv.2021.151760

[16] Liu, W., Wang, P., Meng, Y., Zhao C., and Zhang Z. (2020). Cloud spot instance price prediction using *k*NN regression. *Hum. Cent. Comput. Inf. Sci. 10*, 34. https://doi.org/10.1186/s13673-020-00239-5

[17] Paryudi, I. 2019. What affects k value selection In K-nearest neighbor? *Int. J. Sci. Technol. Res., 8*(7) 86-92. https://www.ijstr.org/research-paper-publishing.php?month=july2019

[18] Kataria, A., Singh, M. (2013). A review of data classification using k-nearest neighbour algorithm. *Int. J. Emerg. Technol. Adv. Eng. 3*(6), 354–360. https://www.ijetae.com/files/Volume3Issue6/IJETAE_0613_60.pdf

[19] Van de Velden, M., D'Enza, A. I., Markos, A., & Cavicchia, C. (2024). A general framework for implementing distances for categorical variables. *Pattern Recognition*, *153*, 110547. https://doi.org/10.1016/j.patcog.2024.110547

[20] Tuerhong, G., Kim, S.B. (2014). Gower distance-based multivariate control charts for a mixture of continuous and categorical variables. *Expert Syst. Appl., 41*(4), 1701–1707. https://doi.org/10.1016/j.eswa.2013.08.068.

[21] Sulc, Z., Procházka, J., and Matějkaz, M. (2016). Modifications of the Gower similarity coefficient. The Proceeding of 19th Appl. Math. Stat. Econ. 2016; Banská Štiavnica, Slovakia; Matej Bel University [Online]. https://www.researchgate.net/publication/313387106.

[22] Van de Velden, M., D'Enza, A. I., Markos, A., & Cavicchia, C. (2024). *Unbiased mixed variables distance*. *arXiv preprint arXiv:2411.00429*. https://arxiv.org/abs/2411.00429

[23] Kadhim, M.N, Al-Shammary, D., & Sufi, F. (2024). A novel voice classification based on Gower distance for Parkinson disease detection. *International Journal of Medical Informatics*, *191*, 105583. https://doi.org/10.1016/j.ijmedinf.2024.10558

[24] Coombes, C. E., Liu, X., Abrams, Z. B., Coombes, K. R., & Brock, G. (2021). Simulation-derived best practices for clustering clinical data. *Journal of Biomedical Informatics*, *118*, 103788. https://doi.org/10.1016/j.jbi.2021.103788

[25] Yozza. H., Azizah, N.M., Yulianti, L., and Rahmi, I. (2023). The classification of "Program Sembako" recipients in Payobasung West Sumatra based on k-nearest neighbor classifier. *Jurnal Natural* (in Bahasa). 23(2), 83-91. https://doi.org/ 10.24815/jn.v23i2.29738

[26] Yasser, M. (2021). *Song popularity dataset*. Available at https://www.kaggle.com/datasets/yasserh/song-popularity-dataset/data

[27] Araujo, V.S., Cristo, M.A.P., & Giusti, R. (2020). Predicting music popularity on streaming platform. *Revista de Inform*, *27*(04), 108-117. http://dx.doi.org/10.22456/2175-2745.107021

[28] Van de Velden, M., D'Enza, A.I., & Markos, A. (2019). Distance-based clustering of mixed data. *Wiley Interdisciplinary Reviews: Computational Statistics*, *11*(3), e1456. DOI: 10.1002/wics.1456

[29] Kumbure, M.M., & Luukka, P. (2022). A generalized fuzzy *k*-nearest neighbor regression model based on Minkowski distance. *Granul. Comput, 7*, 657–671. https://doi.org/10.1007/s41066-021-00288-w

[30] Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings, 3*(1), 91-99. https://doi.org/10.1016/j.gltp.2022.04.020

[31] Nijkamp, R. (2018). *Prediction of product success: explaining song popularity by audio features from Spotify data* [paper presentation]. 11th IBA Thesis Conference, University of Twente, Enschede, The Netherlands

[32] Jamdar, A., Abraham, J., Khanna, K., & Dubey, R. (2015). Emotion analysis of songs based on lyrical and audio features. *Int. J. Artif. Intell. Appl*., *6*(3), 35–50. https://doi.org/10.5121/ijaia.2015.6304

[33] Kowald, D., Schedl, M., & Lex, E. (2019). The unfairness of popularity bias in music recommendation: A reproducibility study. *arXiv preprint arXiv:1912.04696*. https://doi.org/10.48550/arXiv.1912.04696