Machine Learning Methods for the Prediction of Scores in the *Eurovision Song Contest*

Davor Penzar
freelancing data science and machine learning enthusiast
Zagreb, Croatia
davor.penzar@gmail.com

***Abstract*—This paper introduces a few machine learning methods and possible models for score prediction in the *Eurovision Song Contest*. The idea is to train a model for predicting mainstream taste in music, which could be useful in music industry if done successfully. To achieve that, the sound is processed using deep convolutional networks, some of which take in extra non-auditory song features. After the analysis of the results, the models are employed to predict the score of the Croatian candidates for the 2020 edition of the Contest had it been held.**

***Index Terms*—music information retrieval, computer audition, deep learning, machine learning, convolutional neural networks, *Eurovision Song Contest*, score prediction, regression, classification**

I. Introduction

There is no doubt the majority of a *logically* grouped—such as dividing by age, mother tongue, level of education etc.—part of the population share some commonalities in their taste in various forms of art. Naturally, such is the case in music. To illustrate this, in a world where each individual would have their own taste in music completely independent of the others', the popularity of songs would be distributed relatively uniformly. This is obviously not the case: for example, one may observe scores and reviews of songs at [2], *WoC* of top charts at [24] or views of trending music videos at [38]—simultaneously noticeable, on the other hand, is the great domination of just a few music genres[1]. Obviously, the sources mentioned are updated frequently, but it is no surprise that the findings are always the same. In fact, even more stable statistics, such as those at [5], lead to analogous conclusions. After all, music industry would probably not be as profitable as it is (v. [33]) if the *common taste* did not exist.

The phenomenon is not a novelty, either. One could read some interesting facts about the 19th century pianist virtuoso and composer Franz Liszt's popularity at [35] or enjoy the 1984 M. Forman's and P. Shaffer's movie *Amadeus* displaying, although dramatically and comically amended, the glorious life and tragic death of the famous 18th century composer Wolfgang Amadeus Mozart. Surely we may rely on the idea that the phenomenon will continue on in the future as well.

What makes a song (a musically-lyrical piece) a *hit* could be analysed through music theory, literature, linguistics, sociology and other scientific aspects, but that will not be done in this paper. Rather, the idea of the paper is to train an *alien*, completely oblivious of the aforementioned disciplines and their points of interest in songs and pop culture, to predict how the public would respond to a music piece just by hearing it. Of course, such an ignorant is impossible to find (probably a literal intelligent, sound hearing extraterrestrial alien would be the most applicable candidate), therefore compromises had to be made. Due to the area of knowledge and interest of the author, a machine learning model was chosen as a representation of the *alien* mentioned above.

If a model proves itself successful, it might be used in music industry to filter out most promising song writers, authors, performers, records and albums, ultimately to boost a record label's business. Not only could this be done faster than by a human, but the process would also be cheaper. Some may argue against this; nevertheless the paper should primarily be viewed as an intellectual exercise in music information retrieval, computer audition and

[1] Similar statistics could be found for books at [14], for movies and TV shows at [17, 30]…

other close fields. However, the author's personal point of view on the possible repercussions of employment of the models in actual music industry is given in the conclusion, subsection VI-A.

A. *Related Works*

As stated in a popular science article at [12], *Spotify* predicts music taste by comparing the user's favourite songs with other users' playlists (k nearest neighbours or a similar, more complex model). Successful or not, the method does not operate on actual auditory features of songs that the final user enjoys or dislikes but it assumes the same *common taste* on which this paper is based. On the other hand, authors at [26] correlated audio features to individuals' preferences in songs, which is very close to the idea of this project. Similar research was conducted at [22]. Another popular science article at [31] states that not only could a machine learn artistic aesthetics[2], but it could also generate art. The article then references an interesting music analysis and composition model proposed at [21]. On the other hand, visual aesthetics prediction seems to have been developed more thoroughly as seen, for example, at [19, 20].

However, machine learning has been extensively employed in audio processing and music information retrieval for classification and tagging purposes: e. g. at [6, 15, 16, 18, 27, 28]. Nonetheless, the author of this paper strongly believes that tags and other objective classes observed are related to listeners' taste and preferences. As a result, the models explained in the aforementioned sources were studied for the purpose of this project.

Concerning the *Eurovision Song Contest*, artificial intelligence and machine learning, the connection has already been made. In 2019 a song has been created by a machine after learning from a number of *Eurovision* songs, as described at [1, 7, 32]. A year later, in 2020, an artificial intelligence song contest was held, available at [36, 37]. The contest was even announced by *Eurovision* at [9].

B. *The Choice of the* Eurovision Song Contest

The *Eurovision Song Contest* (*ESC*) was chosen as the measurement of the popularity of songs because of its 6 decades long continuous history (except for the 2020 edition due to the pandemic—v. [10, 11]) and the easily understood scoring system. Unlike measuring popularity of songs by assigning scores to their positions in top charts, numbers of sales and other implicit indicators, the scoring system of the *ESC* already displays numerically valued popularity of the competing songs expressed through the audience's and the jury's votes. Also, all songs in a single edition of the Contest had been introduced roughly at the same time and had been available to be listened to approximately the same amount of time, meaning all of them had somewhat equal chance of having their popularity faded out by the time of the contest. Finally, local music stars are not necessarily popular abroad, hence songs' scores might be—and, for the purpose of this paper, they hopefully are—independent of artists' popularity.

On the other hand, by checking individual scores at [25] (or by watching the *ESC* over the years) one could notice that some neighbouring or mutually friendly countries usually give each other high points, probably regardless of their actual enjoyment of contestants (songs). Such bias disrupts the idea that the popularity of a song could be predicted merely by analysing how it sounds, indicating that some additional features might have to be introduced. Besides the country of origin of the song, the features may include the language in which the lyrics are, the category of the performer (a solo singer or a band), the performer's sex (or the sex of the lead singer in case of a band) and other easily deducible information. In the end, the author believes that the number of voting countries and the fact that all countries' votes are of equal weights *smooth out* the noise in scores.

Of all the songs competing in the *ESC*, only songs from years 1957–2019 were used because, as mentioned at [11], voting was secret in 1956 and the scores were not published, plus, in the year 2020 the Contest was not held. From the years observed, only the songs and scores from finals (if semifinal round(s) existed) were used because semifinals' and finals' scores are, obviously, not comparable.

[2] In the opinion of this paper's author, art is not solely about aesthetics, and sometimes it is not about aesthetics at all—it all depends on the art form, style and the artist's idea.

II. Dataset

Unfortunately, a complete dataset did not exist beforehand and had to be created manually. The main resources were:

- songs from *YouTube*, most notably from [8],
- scores by countries from [11, 13, 25],
- extra features from [11, 13] and *Wikipedia*.

It is obvious that the acquired dataset is not verifiable and it may not be completely correct. Also, due to legal reasons—unauthorised distribution of copyrighted music—the author is not able to share their dataset with others. Even the *transformed* dataset (extracted features used as the input of the models, split into *windows* of a consistent format/shape, along with extra features and scores) would be difficult to share because of its size measuring in tens of gibibytes (GiB).

A. Regional Division of the Contestant Countries

The country of origin as a non-auditory feature of songs seems to be *too granular* and *too sparse* as a country might have competed in the Contest only once in a decade. As a result, if the country has never appeared in the training dataset of a model depending on the feature, its score cannot be predicted by the model. To avoid such problems, countries may be grouped into regions according to some geographical, political, linguistic and cultural similarities or differences. While partitioning, regions defined by sources at [4, 29] were studied, but the final regions are not perfectly aligned with either of the two divisions.

European countries appearing in the dataset are divided into 6 regions which, according to geographical locations, may be named as:

- Central Europe: Austria, Czechia, Germany, Hungary, Poland, Slovakia,
- Northern Europe: Denmark, Finland, Iceland, Norway, Sweden,
- Western Europe: Belgium, France, Ireland, Luxembourg, Monaco, Netherlands, Switzerland, United Kingdom,
- Southern Europe: Cyprus, Greece, Italy, Malta, Portugal, San Marino, Spain,
- South-Eastern Europe: Albania, Bosnia and Herzegovina, Bulgaria, Croatia, FYR Macedonia, Montenegro, North Macedonia, Romania, Serbia, Serbia and Montenegro, Slovenia, Yugoslavia,
- Eastern Europe: Belarus, Estonia, Latvia, Lithuania, Moldova, Russia, Ukraine.

Additionally, non-European countries (or those not always considered European) are grouped into their own, seventh group: Armenia, Azerbaijan, Georgia, Israel, Morocco, Turkey. Such division is displayed in figure 1 (some European countries not appearing in the dataset are also assigned to a region and coloured for the completeness of the map).

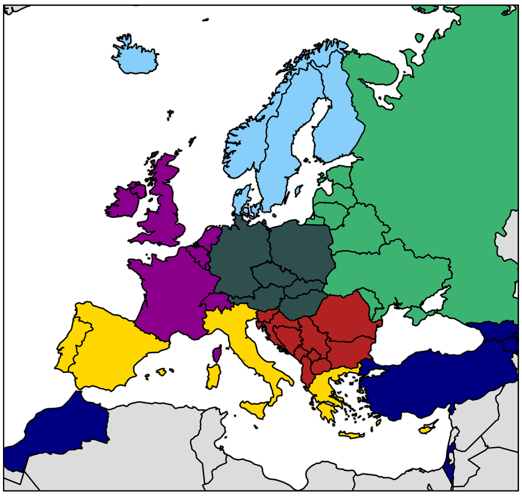

Figure 1: Regional division of Europe used in the project:

- ⬛ Central Europe
- 🟦 Northern Europe
- 🟪 Western Europe
- 🟨 Southern Europe
- 🟥 South-Eastern Europe
- 🟩 Eastern Europe
- ■ Other

Countries coloured lightly grey are not assigned to any region/group (they are irrelevant for the dataset).

Although Australia is clearly a geographical outsider, it is assigned to Western Europe region because of the economic, political, cultural and linguistic relations to other countries from the region, as opposed to countries from the *non-European* group of countries. Similarly, Slovenia and Croatia were the most questionable countries amongst the European countries. They may be assigned to Central Europe region because of historic and cultural affiliations to rulers from the Central and the Southern Europe[3], as well as the predominant Roman Catholic religion, while other countries in the South-Eastern region have stronger eastern and south-eastern influences (e. g. Russia, Ottoman Empire). Impacts on culture and religion are certainly present in folklore and ultimately in popular music, but a stronger coherence in terms of the Contest appears to be between Croatia, Slovenia and other countries from the chosen region when observing voting results at [11, 13, 25]. After all, the main purpose of the feature is to overcome possible noise in scores caused by biased votes.

B. Sound Preprocessing

The preprocessing of raw input songs was done simultaneously with the construction of the dataset. Naturally, songs were originally saved as audio files on a computer. A series of *Python* programs was then run to extract the dataset from the files.

Each song passed a few steps first:

1) it was loaded as a *NumPy* `ndarray` using *libROSA* `load` function as a mono audio signal in a predefined fixed sample rate,
2) zero-crossing rate (ZCR) was computed from the audio signal using *libROSA* `feature.zero_crossing_rate` function,
3) the audio signal (time series) was split into harmonic and percussive components using *libROSA* `effects.hpss` function,
4) constant-Q chromagram was computed from the harmonic component using *libROSA* `feature.chroma_ctq` function,
5) tempogram was computed from the percussive component using *libROSA* `feature.tempogram` function,
6) mel-frequency cepstrum (MFC) was computed from the original audio signal using *libROSA* `feature.mfcc` function,
7) the 1st and the 2nd derivative Δ-features were computed from the MFC using *libROSA* `feature.delta` function.

Of all the data mentioned above, only ZCR, chromagram, tempogram, MFC, Δ-MFC and Δ^2-MFC features were used in the rest of the program. More about the features and their extraction is available at [23].

After the features were computed, the first and last 10 seconds from each song were disregarded (cut out). The reason for this is to avoid using *intros* and *outros* of songs for score prediction since these parts can be very monotone and generic even if middle parts, such as the chorus, are not. Also, a constant signal of magnitude 0—or even the noise from the live performance (e. g. applause)—might be present at the very beginning or the end of an audio file due to imprecise trimming[4]. All possible 10 second long excerpts (*windows*) of the rest of the song were then extracted[5], and *the most diverse sample* of size n of them was used. The number $n \in \mathbb{N}$, $n > 1$, was fixed and the same for all songs, and *the most diverse sample* was found by conducting the process explained in the appendix, subsection VII-A, on mel-frequency cepstral coefficient (MFCC) matrices.

The choice of the most diverse subsample was made to ensure all different parts of each song were included, and that they were all included in the same extent. Moreover, sampling was done in such a way to maximise diversity (variance) of all observations within a single year of competition. Thus, ideally, each song provided something *new*, something different from other songs to make it easier for the model to distinguish why exactly it should receive the score it received.

All hyperparameters (e. g. sample rate, number of MFCCs...) were the same for all songs, resulting in a consistent shape of features, regardless of the song from which the windows originated. Furthermore, as the number n mentioned above was the same for all songs, each song was represented by the same number of observations (windows) no matter what its original duration had been. The final audio

[3] Of course, in the most recent period both of the countries were parts of Yugoslavia—the kingdom and the socialist federal republic—but for a great part of the second millennium the border of the Ottoman Empire stayed just south of Croatian territories.

[4] In fact, the author later discovered that some audio files had even longer non-musical beginnings and ends. However, this was taken as an acceptable risk because manual correction of 1308 audio files was considered too much painstaking work for a single volunteering author, and cutting more than 10 s seemed as discarding too much valuable information from the properly trimmed samples.

[5] In reality there are infinitely many 10 second windows (or, at least, $3 \cdot 60\,\mathrm{s}/t_{\mathrm{P}} \approx 32{,}3 \cdot 10^{45}$ windows in a 3 minute song, where t_{P} is the Planck time), but only discrete time points corresponding to the time points of columns of the chromagram, tempogram and MFCC matrices were used.

dataset was then composed from all the ZCRs, chromagrams, tempograms, MFCs, Δ-MFCs and Δ^2-MFCs on the chosen windows from all songs. Also, for each window in the final dataset the number $p \in [0, 1]$, corresponding to the window's middle point's relative position (beginning of the window plus 5 seconds) in the song, acknowledging the 20 seconds cut out from the beginning and the end of the song, was saved.

C. Splitting the Dataset into Trainig and Validation Datasets

As the title of the subsection suggest, no testing dataset was used—at least no explicit one. This shall be explained later, but a model for predicting scores in the year y is trained and validated on observations from years preceding the year y, i. e. on years $y - 1, y - 2, \ldots, y - n, \ldots$, and then, after being fully constructed (trained), it is tested on all observations in the year y. The training and validation datasets, on the other hand, were constructed for each year independently of other years.

To split the dataset in a given year into a training and a validation dataset, the observations from the year were split into parts of sizes roughly $75\,\% : 25\,\%$ of the original (complete) year's dataset; the training set was then chosen as the larger one. However, to avoid the possibility of a classification model *sneaking into* regression models, this was done by splitting observations (windows of songs) in such a manner that each song is completely in either the training or the validation dataset. Otherwise, if a song's windows were simultaneously in the training and the validation datasets, a model could unintentionally be trained to recognise the song from the features and output its score, without modelling a real and useful regression between the features and the score.

In order to optimally split observations in each year, the objective was to keep the diversity of observations in the training and validation datasets from the original (complete) year's dataset. Similarly to the generation of windows, the diversity of MFCs was observed; however, this time the diversity (variance) of scores was observed as well. *To keep* the diversity means to minimise the absolute difference between the subsample's diversity and the original sample's diversity; here there were two differences to minimise (the training and the validation datasets'). The process of finding the optimal split is explained in the appendix, subsection VII-A.

III. Exploratory Analysis

The exploratory analysis shall only be focused on scores, quantities and auditory features. Non-auditory features are considered as merely auxiliary information, and not something on which one should focus when studying the problem of the paper. The dataset included information from 62 (finals) editions of the Contest (years 1957–2019) with 52 contestants-countries, by differentiating countries such as Yugoslavia from Serbia and Montenegro or F. Y. R. Macednoina from North Macedonia, resulting in total of 1308 entries. After extracting multiple windows from each competing song, the number of elements in the dataset multiplied.

Note that the exploratory analysis was done on the complete dataset, not only on the training part. Normally this would be considered bad practice, but the author believes it is not the case here. First of all, normalisation of scores, as shown in figures 4 and 5, is possible without knowing the actual scores since the average score is calculable just by knowing the rules of voting and the number of voting countries. This is comparable to a situation where data is given in various monetary currencies (v. subsection III-A to understand why): in order to make the dataset useful, a set of unifying formulas must be available, either explicitly—as a currency conversion table (average scores per year in our case)—or implicitly—as a set of features on which the conversion factors are dependant or to which they are correlated (voting rules and the number of voting countries in our case). Also, if both the training and the validation datasets were constructed as representative samples consistently over the years, all graphs from subsections III-A and III-B would remain visually similar, albeit with values potentially scaled down (e. g. the number of contestants). Second of all, examples of extracted features are shown in figures 7 to 13 merely to demonstrate how a raw audio is transformed into features for the models' input. It is actually irrelevant if the example is a part of a training set, a validation set, a testing set or unused by models at all—the transformation is possible on any time series (audio or not) of the correct format.

A. Scores

Let us observe the distribution of scores over the years. The meaning of line colours in figures 2 to 4 (their legend) is the following:

- 🟦 the mean and the surrounding symmetric interval of 2 standard deviations (mean ± sd)[6],
- 🟥 the minimum,
- 🟧 the lower quartile,
- 🟨 the median,
- 🟩 the upper quartile,
- 🟩 the maximum, i. e. the winners' scores.

All of the figures mentioned above, as well as figure 6, display discrete data (one point per year), but points are connected with straight lines to display progression of values.

As one could see in figure 2, scores were much lower in the beginning of the Contest. They were even lower in the early 2000s compared to the scores from the 2010s.

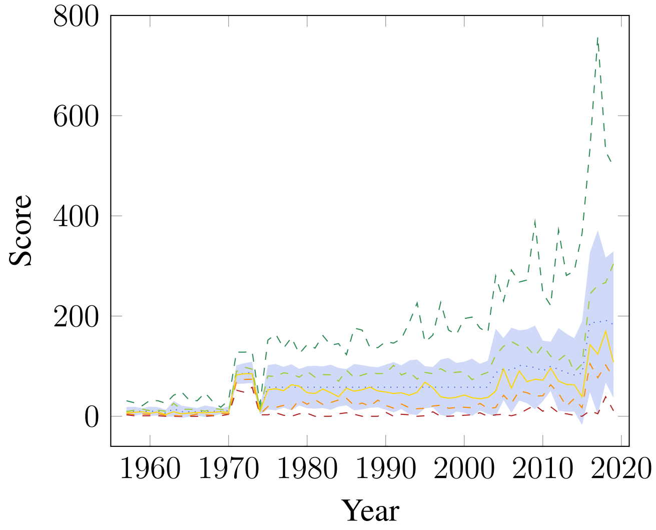

Figure 2: Distribution of scores over years

Of course, one of the reasons why the scores increased over the years may be the increase in the number of contestants, as seen in figure 6. Given a fixed set of rules regarding voting and scoring (independent of the number of contestants), one could expect the scores to be proportional to the number of contestants—after all, by accepting the original idea that the preferability of a song may be predicted, this comes as a natural conclusion (a common taste should exist amongst voters). However, as seen in figure 3, which displays the progression of ratios of the score and the number of contestants over the years, the shape of curves is very similar to those in figure 2—even the measures of central tendency (the mean and the median) are inconsistent. In other words, irregularity of scores' meanings is still present.

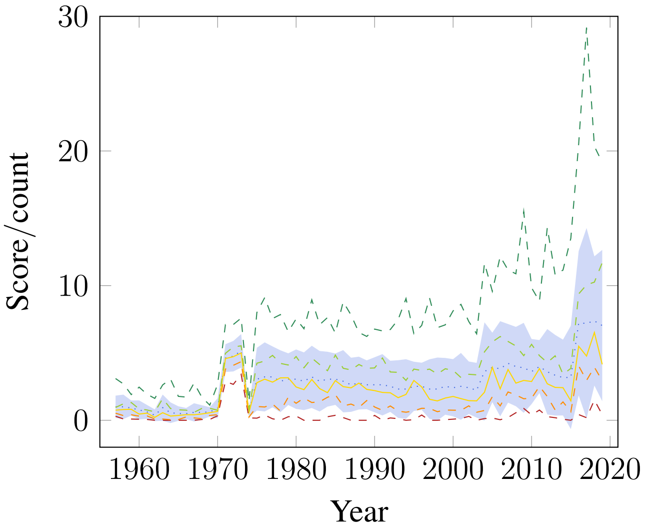

Figure 3: Distribution of scores by count over years

One could standardise scores by subtracting the mean and dividing by the standard deviation ($X \mapsto (X - \mathbb{E}[X]) / \sqrt{\mathit{Var}(X)}$), but such values would be *even more meaningless* than the original scores: for instance, ratios amongst scores of contestants in the same year would be lost. Also, by predicting such values one could only determine the final ranking list, but not the actual scores. Alternatively, the normalisation of scores could be done by dividing scores by the mean in their respective years (divide scores in the year y by the mean of all scores in the

[6] The interval does not have any significant meaning, such as a confidence interval or anything similar. It only shows the magnitude of the standard deviation, implying dispersion of data (scores)—the wider the area, the more dispersed the data is. Note that the width of the area is observed vertically, in the direction of the y-axis, and not perpendicularly to the mean line.

year y)[7]. As seen in figure 4, this produces much more consistent scoring over the years compared to the previous two graphs, by simultaneously keeping ratios amongst scores in the same year. The meaning of normalised scores may also be considered consistent: scoring 2 normalised points means the same regardless of the year—except in the early 1970s when scores were much more densely distributed, probably because of rules explained at [11]. Also, predicting such values is interpretable: by knowing the rules regarding voting and scoring, as well as the number of voting countries in a given year, one could easily calculate the actual predicted score of a song from the predicted normalised score.

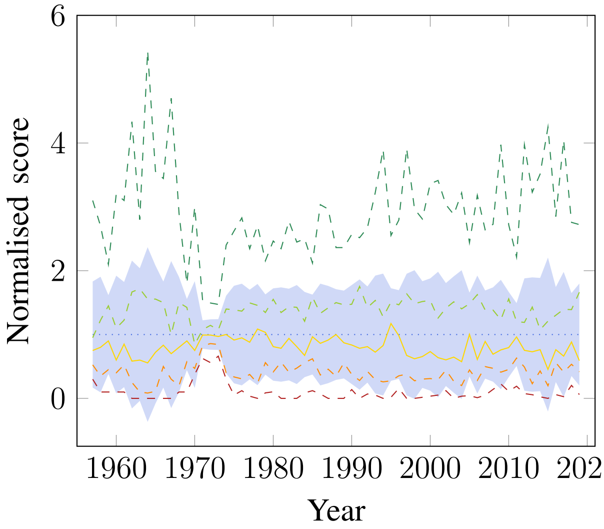

Figure 4: Distribution of normalised scores over years

As seen in figure 4, normalised scores seem to be denser in the lower regions, while the maximum is highly dispersed over the years. By interpreting them as continuous values rather than discrete (computer science, information science, politics, geography, history, biology and physics aside, the number of contestants could be arbitrarily large which could result in an arbitrarily fine distribution of normalised scores), their histogram is given in figure 5. Indeed, the lower the normalised score, the more common it is. The distribution may also suggest that people (voters) from multiple countries share a common taste in music since winners and runner-ups are always voted more or less unanimously.

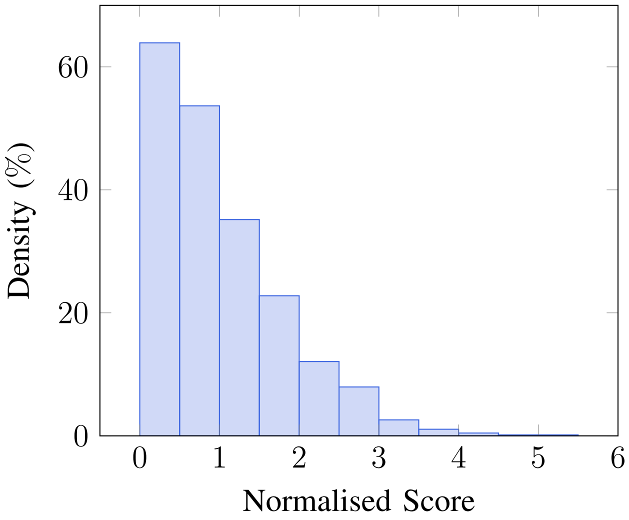

Figure 5: Histogram of normalised scores—all years

B. Number of Contestants

By reading the previous exploratory analysis, it is obvious that the number of contestants has not been constant throughout the years—it ranges from 10 all the way up to 27. Its progression is displayed in figure 6. The average (mean) number of contestants is somewhere between 20 and 21, but the median is exactly 22.

To demonstrate real-life applicability of the models, their success should be observed more in the later years than in the earlier—as Heraclitus said, *everything moves and nothing stays still*; the same is with music trends. Fortunately, on average there was a greater number of contestants in the more recent years making the dataset *richer* as years increase. For example, starting with the year 1987, the number of contestants has always been at least 21, which makes it a continuous period of the most recent 32 years—more than half of the period covered by the dataset.

C. Auditory Features

Finally, auditory features—the key features for the project—are observed. All of them shall be visualised on the same window: a 10 second window starting at about 16,56 s into the Ukrainian's winning

[7] Since the resulting values are not necessarily limited to the interval $[0, 1]$, the term *normalisation* may not be entirely correct. However, as the mean is set to a constant value of 1 by scaling, this term was chosen to describe the transformation at hand.

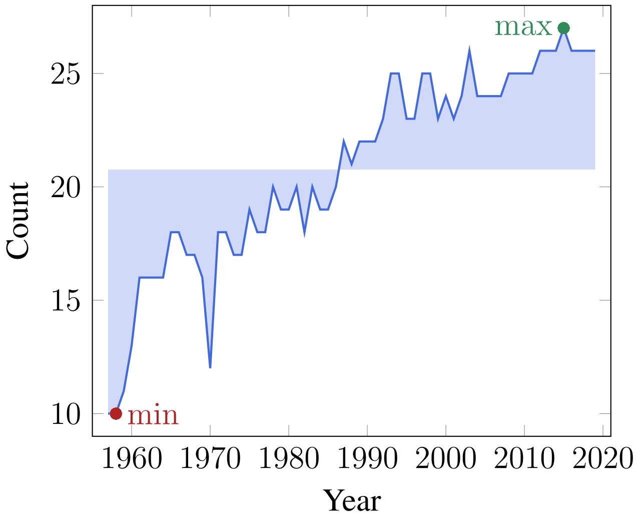

Figure 6: Number of contestants over years. The blue line represents the number, while the lightly shaded blue area represents its oriented distance from the mean

song of the 2004 edition of the Contest, *Wild Dances* (Дикі танці) by Ruslana.

Initially, a raw audio signal time series is given. The series is visualised in figure 7. It was chosen not to operate on such raw information since many explicitly calculable features might not be *found* by a machine learning model, although they could be rationally identified with a human's impression of a song.

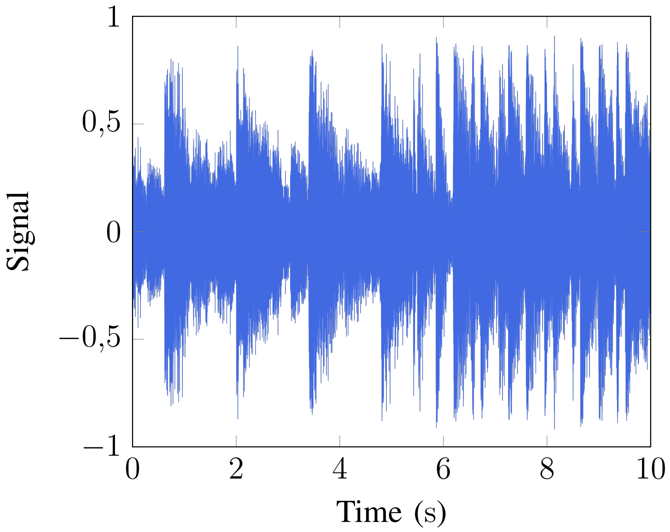

Figure 7: Audio time series visualisation

The *simplest* feature observed is the *zero-crossing rate* of a signal, displayed in figure 8. The feature indicates how many times the signal crosses the value of 0, or how many times it changes the sign. ZCR might seem to be *primitive* as well (as the original signal), thereby being insufficient for the problem at hand.

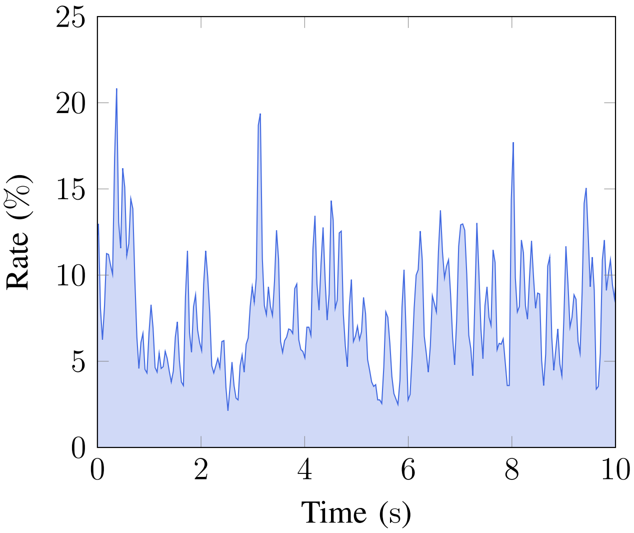

Figure 8: Zero-crossing rate visualisation. Note that the y-axis stretches only up to $25\,\%$

The main feature observed is the *mel-frequency cepstrum* (MFC), displayed in figure 10. In short, it represents a power spectrum of a sound, but a more thorough explanation is available at [23]. The interpretation of the MFC in the context of this paper is *what a human listener hears*. The MFC is computed from a spectrogram on a non-linear mel scale adjusted for the humans' perception of sound frequencies in melodies[8] and can therefore be identified with humans' experience of a musical sound. Although *raw* spectrograms were not used in the project, to further illustrate how an MFC is obtained, the spectrogram of the window is visualised in figure 9. In the figure, the greener the area, the higher the frequency's pressure is at the time point; contrarily, the redder the area, the lower the pressure is. Along the ordinary MFC, its 1st and 2nd derivatives are observed. This is because a human does not (only) hear sounds at isolated time points

[8] Even the *mel* in terms *mel scale*, *mel-scaled spectrogram* and *mel-frequency cepstrum* derives from the word *melody*.

when they listen to music, but they also experience the progression of sounds, while more profound listeners—for instance, those musically educated—even ponder upon *progression of progression*, i. e. how the dynamic elements of music change over time.

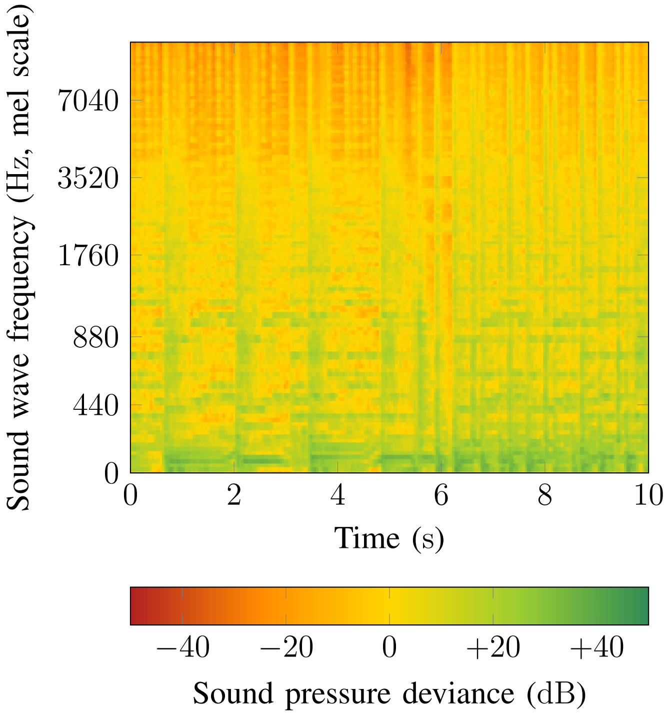

Figure 9: Mel-scaled spectrogram visualisation. Sound pressure deviance is measured against the median at the corresponding time point

Since it is more of a timbral feature, the MFC (implicitly) includes, along others, both rhythmic and melodic elements of the music. A more suitable representation of only melodic and harmonic elements would be a chromagram, displayed in figure 11. It shows intensity of each pitch class, regardless of the octave, through the progression of time.

Originally, at each time point the maximal intensity is 1 (columns of the matrix are normalised using the $+\infty$-norm). The reason for this is that raw melody does not include dynamics (*piano*, *forte* etc.) and that parallel polyphonic progressions should not cancel each other out, which would happen if the standard Euclidean 2-norm was used. For instance, if exactly 2 voices played at the same time equally intense, using the Euclidean norm each of them would have intensity $\sqrt{2}/2$; however, using the $+\infty$-norm they would both have intensity 1, as much as each of them would have if they played solo. On the other hand, possible pauses in music could cause problems: normalising a column of zeros would lead to division by 0 (regardless of the norm used). Therefore for those time intervals in which the original audio signal's magnitude does not exceed a certain threshold the chromagram's values are all set to 0.

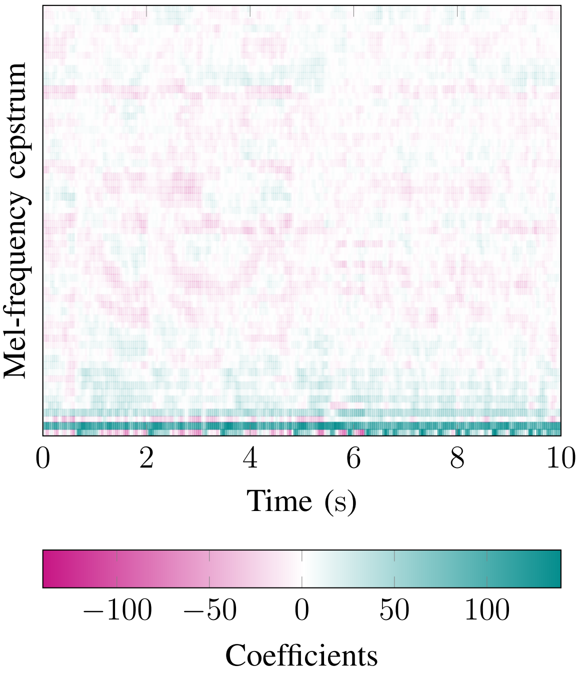

Figure 10: Mel-frequency cepstrum visualisation. No ticks or values are displayed on the y-axis since indices of coefficients, unlike their relative positioning and ordering, are not very informational. For those interested, 64 coefficients represent each time point

Besides normalisation, the *resolution* of a chromagram—the number of columns in the matrix per a unit of time—has to be adjusted as well. Chromagrams in this project were constructed so that each 10 second window was represented by a matrix of 64 columns (the resolution was 6.4 columns per second). For a song in an *allegro* tempo of 128 beats per minute (BPM) this results in 3 columns per beat; for a song in a *prestissimo* tempo of 384 BPM the resolution is 1 column per beat. Setting the resolution too low would result in mixing tones from various beats in a single column, even more so given the window may not start at the beginning of a beat. On the other hand, the songs are not played by a single music part of pure tones with perfectly sinusoidal sound waves, but by multiple parts, some of which are vocal, some are instrumental, some may be electronically distroted… Setting the resolution too high could then result in columns indicating nonexistent tones by analysing too short periods of time when constructing the columns.

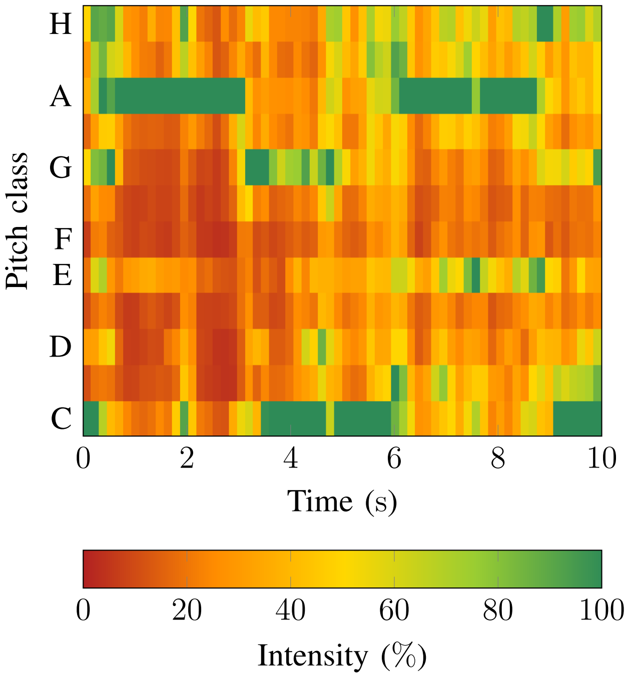

Figure 11: Chromagram visualisation. Note that the 12th pitch is called *H* instead of *B*

Even if the chromagram was constructed optimally in regards to normalisation and resolution, difficulties in music theory are inevitable. First of all, pitches are cyclical (over various octaves). The problem is even visible in figure 11: at the very beginning (around $0\,\mathrm{s}$) and near the end (around $9\,\mathrm{s}$) there seem to be large skips in melody where most intense pitch classes alternate between *C* and *H*. However, a more probable explanation is that the interval was just a step of a semitone (a minor second). Consequently, by displaying each pitch class only once, regardless of the octave, inverted intervals appear the same (the aforementioned *C*–*H* problem is an example of the phenomenon). However, to ensure all semitones are equally distant in the chromagram[9], one could *wrap* the chromagram into the third dimension as illustrated in figure 12. The *wheel* appearing in the figure corresponds to the first column of the chromagram in figure 11. If all columns are transformed analogously and *stacked* together along the z-axis (perpendicular to the x- and y-axes in figure 12—the vector product of their unit vectors), the wheel becomes a cylinder—hence the resulting transformation of the chromagram is called a *cylindrical chromagram*. To display such cylinders as tensors (to input them into prediction models), they must be discretised. Not only does this reduce acccuracy, but overhead memory is used. First of all, the matrix displayed in figure 12 is of dimensions 32×32 with 1024 entries in total—more than 85 times more than the 12 values in the original chromagram column it displays. Second of all, no matter what the dimensions of the matrix are, approximately $(1 - \pi/4)\,N \approx 21\,\%$ of N entries, where $N \in \mathbb{N}_+$ is the total number of entries in the matrix, will remain unused (entries inside the bounding rectangle/square, but outside the relevant ellipse/circle). The excess is then multiplied by the number of time points represented by the chromagram, i. e. the number of columns in the original chromagram.

Another music theory problem of chromagrams—both *ordinary* and cylindrical—is the choice of songs' tuning. Chromagrams used in the project are calibrated for equal temperament tuning system (more about the tuning systems used in modern and historical European music is available at [34]) with A at $440\,\mathrm{Hz}$, although it would probably be correct even if A was tuned anywhere between $428\,\mathrm{Hz}$ and $452\,\mathrm{Hz}$. Tuning A lower than $428\,\mathrm{Hz}$ or higher than $452\,\mathrm{Hz}$ would result in incorrect pitch classes but relative distances between the classes would remain correct[10].

Lastly, the tempogram is observed, displayed in figure 13. Similarly to the chromagram, its columns indicate in which tempo the music is (most probably) played at the given time point. Normalisation rules

[9] Assuming equal temperament tuning system is used, this is preferable. Even if another tuning system is used, semitones do not vary greatly.

[10] In the project's defense, this is no different from people with perfect pitch. Naming tones *C*, *D*, …, *H* is purely theoretical and linguistic.

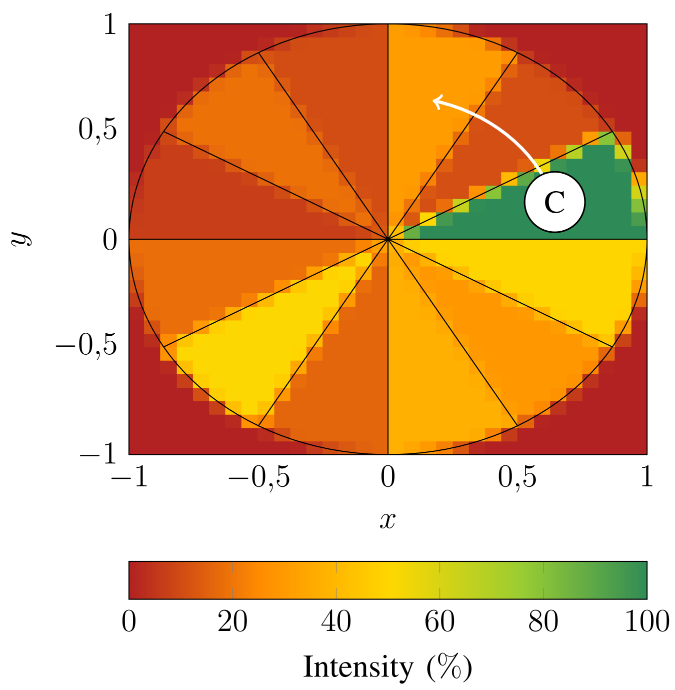

Figure 12: Cylindrical chromagram visualisation. The location of pitch *C* is marked and the arrow originating from it denotes the ascending direction of the scale. Since the cylindrical chromagram is in fact 3-dimensional, only a *slice* of it, corresponding to a single time point, is displayed

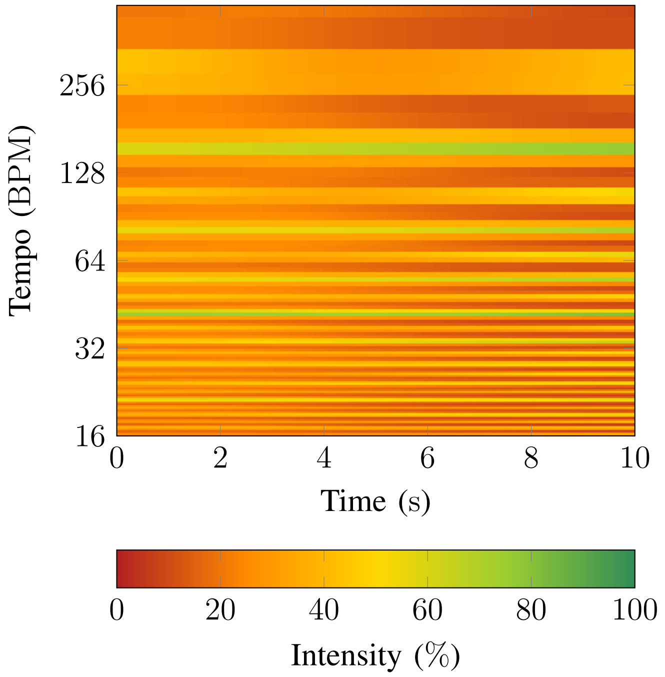

Figure 13: Tempogram visualisation

for tempograms are the same as for chromagrams.

A reasonable assumption is that all songs, or at least the vast majority of songs' windows operated on, should be in a constant tempo. Of course, *rallentando*s and *accelerando*s, as well as instant tempo changes, have existed in music for a very long time (if not forever), but such peculiarities are not so common in modern popular music[11], as they seldom appear more than thrice in a song. Having accepted this, the tempogram matrix may be aggregated into a single vector by averaging all of its rows, creating a mean-vector along its second dimension (time). Again, this vector is normalised using the $+\infty$-norm. Furthermore, as seen in figure 13, more than one tempo is of a high intensity at once making the intensities arranged quite regularly. This is because, for instance, a tempo of 120 BPM may be misinterpreted as a tempo of 60 BPM if every other beat is interpreted as an upbeat. It may also be misinterpreted as a tempo of 40 BPM if only every third beat is interpreted as an actual beat, and the other two are interpreted as parts of a triplet. On the other hand, if actual upbeats are interpreted as true beats, the tempo would be misinterpreted as 240 BPM, and so on. Consequently, the resulting tempogram vector should ultimately be analysed by conducting a discrete Fourier transformation (DFT) for simplification and to circumvent the extra false information. The results of the DFT obtained using *SciPy* `fft.rfft` function are displayed in figure 14. The (discrete) points are connected with straight lines to emphasise the distinction between the real and the imaginary parts as well as to highlight local extrema.

Having inspected some of the DFTs of tempogram vectors from the training dataset, the author of the project concluded that the real part of the DFTs is sufficient enough for a reliable representation. It usually contains 3 distinct and (visually) noticeable *spikes* excluding the first coefficient, which may also be observed in figure 14. The first coefficient is by

[11] The author does not have a fixed reliable source for this information, it is merely the author's opinion. However, it may be noted that the author is not a complete layperson in the field of music theory, as they graduated music theory at a music high school acquiring EQF level 5.

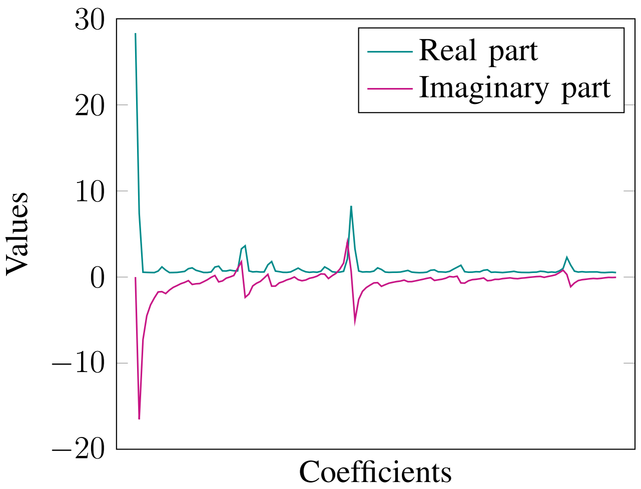

Figure 14: Visualisation of the discrete Fourier transformation of the tempogram. The total number of coefficients is 128

far the largest value in (probably) all DFTs in the dataset, but the DFTs differ in positions of the next three local maxima of the real part. The imaginary part, on the other hand, is highly correlated to the real part and does not seem to provide any additional information[12].

Additionally, the song's duration (in seconds) and the windows' relative positions in songs are also considered auditory features. The reasoning is similar to that in justifying why Δ-MFCs and Δ^2-MFCs are used: music is a temporal form of art—a music piece's duration is something a listener experiences regardless of other potential, non-auditory features (such as the language of lyrics[13] if they exist). This is comparable to a book's or a movie's duration, or a dimension of a pictorial art work or a statue. Similarly, when in the music piece something audible occurs is also important, whether it is at the beginning, in the middle or in the end. Some true non-auditory features were listed in section II.

IV. Hypothesis, Goals and Expectations

Now that both the analysed features and the target variable were presented in detail and are, hopefully, well understood, a little more concrete intentions may be expressed than the ideas stated in the introduction (v. section I). Of course, the main hypothesis of the project is that the impression of a song competing in the *ESC* on the general public demonstrated by the final score could be predicted only from the way the song *sounds*. More precisely, to predict scores of songs competing in the year y to a certain degree of accuracy, a statistical/mathematical model (a computer program) could be developed by analysing the sound and the scores of songs competing in the preceding years. Hopefully, the degree of accuracy would be high enough to at least predict the ranking list, if not the actual scores.

Realistically speaking, absolute scores would be hard to accurately predict using the models proposed in this paper since there is probably a great number of unconsidered variables. For instance, if two relatively equally favourable songs compete in the same year, equally high (or low) points would be split between the two, making each of them score lower than if the other had not competed as well. This may in fact be a part of the reason why lower scores are denser than higher scores (recall figure 5). Another realistic scenario would be a contestant having a great song but a terrible stage performance, or the other way around, in which case the visual impression could impact the score—negatively or positively. The former situation could not be predicted by models proposed in this paper because the idea is to construct a regression that predicts a single (independent) score at once, and not multiple scores of many concurrent contestants. The latter situation goes beyond the scope of this project as only the relationship between the audio and the score is analysed, ignoring the visual components.

The author of the project would consider a model successful if it could identify the top few contestants (songs) within a single year of the Contest and rank them correctly. As discussed in subsection III-A and seen in figures 2 to 5, scores are dense in the lower regions making it more probable that disregarded

[12] The *additional information* mentioned does not mean that the feature is redundant for reconstructing the original values. It means that it may add unnecessary complexity to a model such as a linear regression or a neural network, which may be thought of as an *advancement* of the former. As is well known, inserting linearly correlated features into a linear regression only adds the number of coefficients to compute without improving the results.

[13] Only the purely linguistic *dimensions* of lyrics, such as semantics, are considered non-auditory. Of course, different languages sound differently, but this is covered by the observed features, mostly by the MFCs.

nuances determine the corresponding parts of the ranking list. Furthermore, in real-life business it would matter the most if a program could identify only the most promising songs: if a song is mediocre at best, there is really no need to analyse how dull it actually is, but the available resources could be invested in finding a better alternative. Still, at least a few of the top contestants should be detected by a useful model. Specifically, if only the absolute winner is identified, the model would not seem stable, not even at the top of the ranking list. Thus such a model would not be reliable—the fact that the winner was 'identified' might have merely been a coincidence.

When observing models that take into the account the country of origin or even the region (v. subsection II-A) of the contestant, it must be noted that, as much as the feature could *help* the model in predicting, it could also *handicap* it. This is because such a feature might only balance the noise in the training and the validation scores, both of which are exclusively from the previous years. The composition of contestants (national and regional) may completely change in the year that is to be predicted, altering the way a national/regional relation affects the score. Furthermore, such models could only be validated, tested and, most importantly, employed on songs from countries/regions having appeared in their training datasets. In consequence the model might even *overlook* the winner it was supposed to predict. As in many other situations, a simpler and more general solution achieving similar or better results would be far more preferable.

V. Architecture of Models

Although the idea is—for a given model design (its type and the choice of hyperparameters)—to train a model for each year, a *model* shall actually denote the model architecture (design) throughout this section, unless stated otherwise. That is, the term *model* is an abstraction of all of its instances, each trained on its own training and validation sets with its own prediction goal (the year for which it was trained). This is similar to describing properties and methods of classes in object-oriented programming (OOP), which are common to all objects of the same class, through describing them in a general way for an arbitrary class instance.

Initially the author approached the problem overly optimistically, with an idea to develop custom deep convolutional neural networks (CNNs) *from scratch*. Having been faced with the scarcity of data, it was apparent that this method was destined to be unsuccessful. The number of examples is simply insufficient to train such a complex model. For instance, to train a model for the year y based on the results from the previous 10 years, under the assumption that the average number of contestants per year was 25 (recall figure 6), only 250 songs are available. About a quarter of them would be allocated to the validation dataset, resulting in a training set of only 185–190 songs. Of course, the number of windows extracted from each song may in fact be very large, possibly generating a training dataset of as much as 75k examples (plus additional 25k validation examples for a total of 100k examples), but the resulting windows would be largely overlapped, if not nearly identical. To make matters worse, repetition of elements (such as chorus or bridge, or merely a motif) is common in music, making the extracted windows similar even if they do not actually overlap. In the end, the surplus dataset would not improve training, but would speed up the overfitting of the resulting model.

Even if a model was trained on the complete dataset to predict results in any of the years in the near future (after the final year in the dataset), the dataset would be composed of only 1308 songs. In contrast, the subset of the *Million Songs Dataset* (*MSD*) at [3] is more than $7{,}5$ times larger, while the complete *MSD* is nearly 765 times larger. It is reasonable to assume that if a valid evaluation of songs was defined—e. g. a generalisation of individual *preference* and *confidence variables* from [26]—a model trained on the *MSD* would outmatch the one trained on the given dataset. This is especially the case with complex models such as CNNs.

Nevertheless, *native* CNNs (trained only on the obtained *ESC* dataset) were constructed, mostly for benchmark reasons. Other than them, the constructed models include:

- classifiers – predicting whether or not a song will rank above a certain quantile threshold:
  - logistic regression,
  - support vector machine (SVM),

- (single) decision tree,
- random forest (ensemble of decision trees),

- regressors – predicting the score of a song:
  - linear regression,
  - neural network (NN),
  - CNN (already mentioned).

In addition, each type of a model (e. g. SVM or NN) was trained on top of pretrained models available at [27, 28], through transfer learning (TL). The author of this project expected that the models built upon the pretrained ones would yield better results, simply because of the pretrained models' reported results, richer training datasets and the higher-level features they output (compared to relatively low-level features explained in sections II-B and III-C which were used for the initial models).

VI. Conclusion

A. Significance of Computer-Aided Art Form

Briefly, the author strongly emphasises that the human mind and creativity cannot (or should not) be substituted by a computer in the process of actual art making[14]. Still, mainstream music industry is already largely aided by computers and there is no harm in automating it a little bit more—unless one argues that dehumanisation of work leads to unemployment, social disbalance and other negative consequences.

The reason why a computer is incapable of creating true art does not stem from the author's elitist and overly anthropocentric views, but it is because computers are deterministic machines constructed and programmed by humans. Even when running a *nondeterministic* program, it is in fact deterministic but based on a pseudorandom algorithm and/or an external seed state, therefore true autonomy, creativity and uniqueness needed to *create* art (and not just imitate others) cannot be obtained. If something art-worthy is actually created by a computer, the programmer and/or the initiator of the program should be credited because they are the ones that started and dictated the process of creation, ultimately run by the computer.

VII. Appendix

A. Variance and Standard Deviation of a Multidimensional Sample

Let $d \in \mathbb{N}_+$ and $m_1, m_2, \ldots, m_d \in \mathbb{N}_+$. Suppose $n \in \mathbb{N}$, $n > 1$, observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ of a d-dimensional real-valued population were measured. Let $X \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d \times n}$ be the tensor of the sample such that for every $i = 1, 2, \ldots, n$ its i-th slice along the last dimension is the observation x_i—we shall call this tensor the *sample*.

Let $\bar{x} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ be the mean of the sample X (each position in $\bar{x}$ holds the mean of values at the corresponding positions in observations $x_1, x_2, \ldots, x_n$). Finally, let tensor $X' \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d \times n}$ of joint transformed observations $x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_n - \bar{x}$ be constructed in the same way as the tensor X from the original observations.

The *diversity* of the sample X is then measured by the Frobenius norm of the tensor X'. More precisely, the square of the Frobenius norm divided by an appropriate denominator shall be used. As one can see, when $d = 1$ and $m_1 = 1$, the sample variance is

$$\begin{aligned} \mathit{Var}\left(X\right) &= \frac{1}{n-1}\left(\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2\right) = \\ &= \frac{1}{n-1}\left\|X'\right\|_F^2 . \end{aligned}$$

Furthermore, when $d = 1$ and $m_1 \geq 2$, the trace of the sample covariance matrix actually equals (the notation is explained immediately after the equation)

$$\begin{aligned} \operatorname{tr}\left(\Sigma\left(X\right)\right) &= \sum_{i=1}^{m} \mathit{Cov}\left(X^{(i)}, X^{(i)}\right) = \\ &= \sum_{i=1}^{m} \mathit{Var}\left(X^{(i)}\right) = \\ &= \sum_{i=1}^{m} \frac{1}{n-1}\left(\sum_{j=1}^{n}\left(x_j^{(i)} - \overline{x^{(i)}}\right)^2\right) = \\ &= \frac{1}{n-1}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_j^{(i)} - \overline{x^{(i)}}\right)^2\right) = \\ &= \frac{1}{n-1}\left\|X'\right\|_F^2 , \end{aligned}$$

[14] To be clear, art can be made *using* a computer, but not *by* a computer. Moreover, it is important to notice when and where the computer is used: artistically high valued electronic music compositions have been produced since the introduction of electronics in music, but, in the author's opinion, there would be nothing artistic in an auto-tuned soprano singing *Le Rossignol et la Rose* by Camille Saint-Saëns—at least not without a strong and well articulated idea (such as satire) behind it.

where $m := m_1$ is the number of variables in X (the number of rows), $X^{(i)}$ is the i-th row (variable), $x_j^{(i)}$ is the value of the j-th observation in the i-th row (variable) and $\overline{x^{(i)}}$ is the mean of the i-th row (variable) $X^{(i)}$. Obviously, the appropriate denominator should then be $n - 1$.

Because of the interpretation given above, the value $\frac{1}{n-1} \left\| X' \right\|_F^2$ shall be called the *variance* of the sample X, while its square root shall be called the *standard deviation*. The former shall be denoted $Var\left(X\right)$ and the latter $\operatorname{sd}\left(X\right)$. Similarly to the equations above, one can even prove that the values computed this way actually coincide with the trace and its square root of the covariance matrix[15] of the vectorised observations (*flattened* into vectors of dimension $m_1 m_2 \cdots m_d$)—however, the proof is left as an exercise to the reader. The final values do not depend on the choice of the orthonormal basis, which makes the values even more relevant and significant. For instance, the variance is equal to the trace of the covariance matrix of primary components, which is a diagonal matrix with diagonal elements being exactly variances along said primary components.

Apart from a sample's variance, covariance between two samples may be computed as well. Suppose $Y \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d \times n}$ is another sample of observations $y_1, y_2, \ldots, y_n \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$, $\bar{y} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ is the mean of said observations and $Y' \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d \times n}$ is the analogously transformed sample. Furthermore, let $M_{X'}, M_{Y'} \in \mathbb{R}^{n \times (m_1 m_2 \cdots m_d)}$ be mode-$(d+1)$ unfoldings (over the dimension along which the observations are stacked) of tensors X', Y' respectively. The covariance is then defined as

$$Cov\left(X, Y\right) = \frac{1}{n-1} \operatorname{tr}\left(M_{X'} \cdot M_{Y'}^{\mathsf{T}}\right)$$

($A \cdot B$ denotes the standard matrix multiplication, but the operator $\cdot$ is explicitly marked to avoid confusing $'$ for ,). Since the expression actually evaluates to simply multiplying differences between elements at identical positions and the corresponding means of values, and finally computing the sum of the multiplications, the covariance may actually be generalised to any finite number ($m \geq 2$) of samples. However, such a general case cannot easily be expressed as the trace of a matrix multiplication, but inner products of tensors and other tensor manipulations would have to be introduced.

Of course, to find a (sub)sample of size n amongst N observations with a specific characteristic of its variance (maximal, minimal, as close to the original sample's as possible...)—or, for that mather, (sub)samples with a certain covariance—by brute-force algorithm, the number of samples to inspect equals $\binom{N}{n} \in \mathcal{O}\left(N^{\min\left(\left\{n, N-n\right\}\right)}\right)$, which may be practically too large when $N \gg n \gg 1$. Therefore the (sub)sample may be found by inspecting at most $k \in \mathbb{N}_+$, $k \leq \binom{N}{n}$, samples chosen at random (uniformly) and keeping the optimal amongst them. If the algorithm is done sequentially, it may be optimised even further by letting it stop early if the objective does not improve in a fixed, predetermined number of consequent iterations. Alternatively, an optimisation algorithm may be employed: each consecutive guess is generated by analysing the previous guesses and their objective values.

B. Binary Classification Evaluation Metrics for Ranking Lists of Scores

Let $X = \left\{x_1, x_2, \ldots, x_n\right\}$ be a sample of $n \in \mathbb{N}_+$ observations. Suppose the observations are scored by a scoring function $y \colon X \to \mathbb{R}$. Observations in X may be ranked according to their scores: $x_i \prec x_j$ if and only if $y\left(x_i\right) < y\left(x_j\right)$, for all *legal* indices i, j. Note that the relation is irreflexive and transitive, meaning it defines a partial order over the sample X (the order is total, or linear, if the evaluation y is injective).

Since every scoring of observations in the sample X defines its own partial ordering, notation $\prec$ may be uninformative when comparing ranking lists of two scoring functions. It just does not indicate by which scoring function it has been defined. Thus notation

$$IR\left(X, y\right) := \left\{\left(x_i, x_j\right) \in X^2 : y\left(x_i\right) < y\left(x_j\right)\right\}$$

shall be used instead. The capital letters *IR* represent the word *interrelations*. The set $IR\left(X, y\right)$ is actually the set of all possible hierarchical relationships of observations from the sample X according to the ranking list generated by the scoring y. Analogously,

[15] Still, due to numeric reasons, the values shall be computed as defined rather than from the covariance matrix to reduce computational workload.

the set of interrelations $IR\left(X, y^{*}\right)$ generated by another scoring function y^{*} may be observed.

The alternative notation also allows to concentrate only on interrelations *originating* from a subsample $U \subseteq X$:

$$\begin{aligned} & IR\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(X, y\right) : x_i \in U \vee x_j \in U\right\}, \end{aligned}$$

i. e. $IR\left(U; X, y\right)$ denotes all interrelations $\left(x_i, x_j\right)$ from $IR\left(X, y\right)$ such that at least one of their *ends* is in U. Furthermore, one may be only interested in:

- the *outer* interrelations of U:

$$\begin{aligned} & IR_{\text{out}}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : \right. \\ & \qquad \left. : x_i \notin U \vee x_j \notin U\right\}, \end{aligned}$$

- the *inner* interrelations of U:

$$\begin{aligned} & IR_{\text{in}}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : \right. \\ & \qquad \left. : x_i \in U \wedge x_j \in U\right\}, \end{aligned}$$

- the *right* interrelations of U:

$$\begin{aligned} & IR^{(<)}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : x_i \in U\right\}, \end{aligned}$$

- the *left* interrelations of U:

$$\begin{aligned} & IR^{(>)}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : x_j \in U\right\}, \end{aligned}$$

- the *outer right* interrelations of U:

$$\begin{aligned} & IR^{(<)}_{\text{out}}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : \right. \\ & \qquad \left. : x_i \in U \wedge x_j \notin U\right\}, \end{aligned}$$

  or the *outer left* interrelations of U:

$$\begin{aligned} & IR^{(>)}_{\text{out}}\left(U; X, y\right) \coloneqq \\ & \quad = \left\{\left(x_i, x_j\right) \in IR\left(U; X, y\right) : \right. \\ & \qquad \left. : x_i \notin U \wedge x_j \in U\right\}. \end{aligned}$$

Since *inner* interrelations are simultaneously also *inner right* interrelations and *inner left* interrelations, the two subsets of interrelations are not explicitly mentioned. Reasons why one might observe the mentioned subsets of interrelations may include the need to inspect how observations from U are ranked globally (compared to all observations from X), how they are ranked compared to other observations, how they are ranked internally (compared only to other observations from U), if they are ranked higher or lower in the global ranking list and if they are ranked higher or lower than other observations.

If y is the reference scoring, a scoring y^{*} may be evaluated, amongst others, by metrics used for evaluating (binary) classification models. For instance, $IR\left(X, y\right)$ may be considered the set of *positives* and $IR^{\complement}\left(X, y\right)$ (the set of interrelations $\left(x_i, x_j\right)$, $i \neq j$, such that $\left(x_i, x_j\right) \notin IR\left(X, y\right)$ or, equivalently, $y\left(x_i\right) \not< y\left(x_j\right)$) may be considered the set of *negatives*. Analogously, $IR\left(X, y^{*}\right)$ is then considered the set of predicted *positives* and $IR^{\complement}\left(X, y^{*}\right)$ the set of predicted *negatives*. Given all of these sets, accuracy, precision, recall (sensitivity) and specificity of the scoring y^{*} are all well defined, as well as its F-score.

However, if the total number of observations n is *very large* or if the scoring y actually makes *significant* difference only at the top of the ranking list, the complete ranking list might not be (equally) relevant. A number $k \in \mathbb{N}_{+}$, $k \leq n$, may then be chosen to observe, instead of observing all interrelations from $IR\left(X, y\right)$, only the interrelations from $IR\left(K; X, y\right) \subseteq IR\left(X, y\right)$, where $K \subseteq X$ is the set of the top k observations from X in respect of the reference scoring y. All interrelations in the set of negatives $IR^{\complement}\left(K; X, y\right) \subseteq IR^{\complement}\left(X, y\right)$ (note the $\subseteq$ inclusion instead of the $\supseteq$ inclusion with usual (actual) set complements) is then constituted by all interrelations $\left(x_i, x_j\right) \in X^2$, $i \neq j$, such that at least one of x_i, x_j is in K, but $y\left(x_i\right) \not< y\left(x_j\right)$. Moreover, if, for instance, interrelations $IR_{\text{in}}\left(K; X, y\right)$ were used, both of x_i, x_j would have to be in K for negatives as well as for positives. However, in reality observing only the positives' set $IR_{\text{in}}\left(K; X, y\right)$ would be insufficient because some scoring might internally rank observations from K perfectly, but globally rank them at the very bottom of the ranking list instead of the top.

If $IR\left(X, y\right)$ is a total (linear) order over the sample X, i. e. if the scoring y is injective, both sets of positives vs. negatives are perfectly balanced (they are equipotent meaning they have the same number of elements). This stems from the simple fact that the set of real numbers $\mathbb{R}$ is totally (linearly) ordered. Actually, if the subset of interrelations $IR\left(K; X, y\right)$ is observed as the set of positives instead of the complete set $IR\left(X, y\right)$, then y would only have to distinguish the scores of observations from K, but it may generally be non-injective. More precisely, scores of two distinct observations outside of K could be the same. For instance, if scores are non-negative, more than one irrelevant observation may have a score of 0.

One might wonder why *invention* of new metrics based on the mathematical set theory is required for something that could be evaluated by a *simple* (well-known) Pearson correlation coefficient (PCC) or a Spearman correlation coefficient (SCC), and the corresponding p-value. Between the two coefficients, the SCC would be the more appropriate choice if only the ranking list should be evaluated, and not the linear correlation of scoring functions—the former is also the case with the adaptations of binary classification evaluation metrics defined above. The difference is that the *accuracy* of a ranking list defined by a predicted score naively indicates what the odds are that the underlying scoring function (predictor) will mutually rank any two specimina correctly, while the correlation is more meaningful on a larger number of observations. By transitivity, the accuracy metric may then be interpreted as the accuracy of the predicted ranking list; similar interpretations of other metrics may also be made. On the other hand, the correlation indicates how well a monotone curve may be fitted to the actual and the predicted results. However, for completeness of the analysis, both the binary classification metrics and the correlation coefficient(s) should be observed, and neither should be disregarded.

Of course, if actual scores matter, and not just the ranking list generated by the scoring, then other metrics should also be inspected when evaluating a prediction scoring system. These include, but are not limited to, the metrics for evaluating regression, such as the (root-)mean-square error ((R)MSE) or the mean absolute (percentage) error (MA(P)E).

C. Reconstruction of a Likeable *Song*

The sound analysis conducted in this project starts from a raw audio signal represented as a time series for a chosen sample rate. The sample rate, as well as other parameters used in the analysis, were consistent and independent of the song, year, country of origin or any other feature. Consequently, the results of the analysis are comparable ensuring meaningfulness of the constructed models.

However, there is an additional consequence to this consistency. The time series are effectively real-valued vectors, although not of the same length (dimension, size) due to differences in duration of the songs. Regardless, the complete songs were merely original, raw data, while the actual input for the analysis, and later the prediction, were windows of a consistent duration—in terms of time series, *subvectors* of consistent lengths. These subvectors could then be statistically analysed in order to generate new vectors.

Specifically, the idea is to *reconstruct* audio signals scoring as high as possible at the trained score-prediction models. A *trial-and-error* approach is proposed, i. e. it could be done by constructing a number of different signals, inputting them into the models and choosing the most successful one(s)[16]. Of course, one way to do this is to construct the signals randomly, but, due to a vast number of combinations—many of which would possibly sound to a human like an unpleasant noise and not like a section of an actual song—this does not seem promising. An alternative has already been implied in the previous paragraph, and the idea is to narrow the search by statistically analysing audio signals that are known to sound like *music* (since they were cut out from real songs).

An approach that does not actually operate on raw subvectors of time series is proposed. Rather, discrete Fourier transformation (DFT) is conducted on the series. Again, the reason behind this is to avoid generating random noises because the final audio signal would be generated by applying reverse

[16] Instead of generating many audio signals independently and then testing them all, an optimisation algorithm, such as by using *SciPy* `optimize.minimize` function, could be conducted. Ideally, fewer audio signals should be generated to optimise the score, but additional computation and conditions slow the process, mostly because the ability to test many samples simultaneously (parallelly) is lost.

DFT on the generated coefficients. The assumption is that the coefficients will be generated in a way that they would produce a more pleasant, *music-like* audio signal.

However, not all songs share the key, tempo, instrumentals, vocals and other musical and, consequently, audio elements; but, they do share *musical consistency* (consistency of rhythm, melody, harmonics...). As a result, a dependency amongst the Fourier coefficients is expected. To ease the process of generating music from them, independent coefficients would be more preferable (the reason will be explained later). One possible way—the one that is proposed here—is to conduct independent components analysis (ICA) on the Fourier coefficients; the assumptions needed for the analysis are assumed.

After the independent components are obtained, histograms are computed for each of the component. The process of generating a single component c, with (assumed to be continuous) distribution function $F \colon \mathbb{R} \to [0, 1]$, is then simple:

1) a number $\alpha \in [0, 1]$ is generated (pseudo-randomly uniformly if generating without optimisation),
2) the appropriate bin $B_i = [c_i, c_{i+1})$ (or $[c_i, c_{i+1}]$ if it is the last bin) of the histogram is found for which the inequality $F(c_i) \leq \alpha < F(c_{i+1})$ (or $F(c_i) \leq \alpha \leq F(c_{i+1})$ if B_i is the last bin) is satisfied,
3) the bin B_i is assumed to be uniformly distributed, therefore the value
$$c_{i,i+1}(\alpha) = c_i + \frac{\alpha - F(c_i)}{F(c_{i+1}) - F(c_i)} \in B_i$$
is chosen; alternatively, another interpolation could be used for the approximation of the distribution function F instead of the linear spline, but monotony of the distribution function must be preserved.

If the components are not independent (if dependency exists amongst them), the process explained above, or any other algorithm relying on independently generating components based on their distributions, would not be correct. Instead, conditional distribution would have to be used by taking into consideration the values of the components that have already been generated. Not only does this make the algorithm more complex to explain, but it requires more time and/or memory resources to implement (although, the ICA makes the preprocessing transformation and the reverse transformation more complex).

After all independent components are generated, the audio signal is obtained by reversing the transformation. That is, the generated independent components are transformed back into Fourier transformations, and then the reverse DFT is conducted on them. As a result, *windows* of a non-existing song are obtained. If they score well enough at the score-prediction model, they could be used as inspiration for a real song writer (human or technological).

BIBLIOGRAPHY

[1] G. Ackerman. (2019). Robot sings Eurovision kitsch composed by Oracle AI, Israelis. N. Zivitz, Ed., BNN Bloomberg, [Online]. Available: http://bnnbloomberg.ca/robot-sings-eurovision-kitsch-composed-by-oracle-ai-israelis-1.1259853 (visited on 23/12/2020).

[2] AllMusic, *Record reviews, streaming songs, genres & bands*, Web page, Ann Arbor: NETAKTION, 2020. [Online]. Available: http://allmusic.com/ (visited on 20/12/2020).

[3] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman and P. Lamere, 'The million song dataset', in *Proceedings of the* 12*th* *International Conference on Music Information Retrieval*, A. Klaupuri and C. Leider, Eds., International Society for Music Information Retrieval, Miami, 2011, pp. 591–596.

[4] CIA, *The world factbook*, Web page, Langley: The Federal Government of the United States, 2021. [Online]. Available: http://cia.gov/the-world-factbook (visited on 17/04/2021).

[5] N. Cossar, 50 *best selling studio albums*, Web page, Prestatyn: This Day in Music, 2019. [Online]. Available: http://thisdayinmusic.com/liner-notes/50-best-selling-studio-albums (visited on 18/08/2020).

[6] S. Dielman, P. Brakel and B. Schrauwen, 'Audio-based music classification with a pretrained convolutional network', in *Proceedings of the* 12*th* *International Conference on Music Information Retrieval*, A. Klaupuri and C. Leider, Eds., International Society for Music Information Retrieval, Miami, 2011, pp. 669–674.

[7] J. Drake and J. Abel. (2019). Blue Jeans and Bloody Tears, [Online]. Available: http://blogs.oracle.com/uki/blue-jeans-and-bloody-tears (visited on 23/12/2020).

[8] Eurovision Song Contest, *Eurovision song contest*, Web page, Geneva: European Broadcasting Union, 2020. [Online]. Available: http://youtube.com/user/eurovision (visited on 21/12/2020).

[9] ——, (2020). Introducing the AI song contest!, European Broadcasting Union, [Online]. Available: http://eurovision.tv/story/introducing-the-ai-song-contest (visited on 23/12/2020).

[10] ——, (2020). Rotterdam 2020, European Broadcasting Union, [Online]. Available: http://eurovision.tv/event/rotterdam-2020 (visited on 15/12/2020).

[11] Eurovisionworld, *Eurovision voting & points*, Web page, Eurovisionworld, 2020. [Online]. Available: http://eurovisionworld.com/eurovision (visited on 15/12/2020).

[12] D. Fagella. (2019). AI in taste and art – the current state of machine learning for understanding preferences, Emerj, [Online]. Available: http://emerj.com/editorial-opinion/ai-taste-art-current-state-machine-learning-understanding-preferences (visited on 22/12/2020).

[13] M. Flecht, *Eurovision Song Contest database*, Web page, Köln, 2020. [Online]. Available: http://eschome.net/ (visited on 15/12/2020).

[14] Goodreads, *Meet your next favorite book*, Web page, San Francisco: Amazon, 2020. [Online]. Available: http://goodreads.com/ (visited on 20/12/2020).

[15] P. Hamel and D. Eck, 'Learning features from music audio with deep belief networks', in *Proceedings of the* 11^{th} *International Society for Music Information Retrieval Conference*, J. S. Downie and R. C. Veltkamp, Eds., International Society for Music Information Retrieval, Utrecht, 2010, pp. 339–344.

[16] M. D. Hoffman, D. M. Blei and P. R. Cook, 'Easy as CBA: a simple probabilistic model for tagging music', in *Proceedings of the* 10^{th} *International Society for Music Information Retrieval Conference*, K. Hirata, G. Tzanetakis and K. Yoshii, Eds., International Society for Music Information Retrieval, Kobe, 2009, pp. 369–374.

[17] IMDb, *Ratings, reviews, and where to watch the best movies & TV shows*, Web page, Seattle: Amazon, 2020. [Online]. Available: http://imdb.com/ (visited on 20/12/2020).

[18] H. Lee, Y. Largman, P. T. Pham and A. Y.-T. Ng, 'Unsupervised feature learning for audio classification using convolutional deep belief networks', in *Proceedings of the* 23^{rd} *Annual Conference on Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams and A. Culotta, Eds., Neural Information Processing Systems, vol. 22, Vancouver: Curran Associates, Inc., 2009, pp. 1096–1104.

[19] X. Lu, Z. Lin, H. Jin, J. Yang and J. Ze Wang, 'RAPID: rating pictorial aesthetics using deep learning', in *Proceedings of the* 22^{nd} *ACM International Conference on Multimedia*, K. A. Hua, Y. Rui, R. Steinmetz, A. Hanjalic, A. (Natsev and W. Zhu, Eds., Association for Computing Machinery, Orlando, 2014, pp. 457–466.

[20] G. Malu, R. Sarampudi Bapi and B. Indurkhya, *Learning photography aesthetics with deep CNNs*, 2017. arXiv: 1707.03981 [cs.CV]. (visited on 22/12/2012).

[21] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro and J. Romero, 'A corpus-based hybrid approach to music analysis and composition', in *Proceedings of the* 22^{nd} *AAAI Conference on Artificial Intelligence*, R. C. Holte and A. Howe, Eds., Association for the Advancement of Artificial Intelligence, Vancouver: AAAI Press, 2007, pp. 839–845.

[22] B. McFee, T. Bertin-Mahieux, D. P. W. Ellis and G. R. G. Lanckriet, 'The million song dataset challenge', in *Proceedings of the* 21^{st} *International Conference on World Wide Web*, A. Mille, F. Gandon, J. Misselis, M. Rabinovich and S. Staab, Eds., Association for Computing Machinery, Lyon, 2012, pp. 909–916.

[23] B. McFee *et al.*, 'libROSA: audio and music signal analysis in Python', in *Proceedings of the* 14^{th} *Python in Science Conference*,

K. Huff and J. Bergstra, Eds., SciPy, Austin, 2015, pp. 18–24.

[24] Official Charts, *Official singles chart top* 100, Web page, London: Official Charts Company, 2020. [Online]. Available: http://officialcharts.com/charts/singles-chart (visited on 17/08/2020).

[25] S. Okhuijsen, *Eurovision Song Contest scores* 1975–2019, Web page, Utrecht: Datagraver, 2019. [Online]. Available: http://data.world/datagraver/eurovision-song-contest-scores-1975-2019 (visited on 19/08/2020).

[26] A. van den Oord, S. Dieleman and B. Schrauwen, 'Deep content-based music recommendation', in *Proceedings of the* 27*th* *Annual Conference on Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger, Eds., Neural Information Processing Systems, vol. 26, Lake Tahoe: Curran Associates, Inc., 2013, pp. 2643–2651.

[27] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann and X. Serra, 'End-to-end learning for music audio tagging at scale', in *Proceedings of the* 19*th* *International Society for Music Information Retrieval Conference*, E. Gómez, X. Hu, E. Humphrey and E. Benetos, Eds., International Society for Music Information Retrieval, Paris, 2018, pp. 637–644.

[28] J. Pons and X. Serra, 'musicnn: pretrained convolutional neural networks for music audio tagging', in *Proceedings of the* 20*th* *International Society for Music Information Retrieval Conference*, Late-Breaking/Demo, A. Flexer, G. Peeters, J. Urbano and A. Volk, Eds., International Society for Music Information Retrieval, Delft, 2019.

[29] Publications Office of the European Union, *EuroVoc*, Web page, Luxembourg: European Civil Service, 2021. [Online]. Available: http://op.europa.eu/en/web/eu-vocabularies (visited on 17/04/2021).

[30] Rotten Tomatoes, *Movies | TV shows | movie trailers | reviews*, Web page, Beverly Hills: Fandango Media, 2020. [Online]. Available: http://rottentomatoes.com/ (visited on 20/12/2020).

[31] F. Roza. (2019). Artificial artist: can artificial intelligence create art?, Towards Data Science, [Online]. Available: http://towardsdatascience.com/artificial-artist-can-artificial-intelligence-create-art-d7dd6ed98270 (visited on 22/12/2020).

[32] N. Shapira. (2019). Oracle presents: a Eurovision AI song *Blue Jeans & Bloody Tears*, [Online]. Available: http://nimshap.com/ai_eurovision (visited on 23/12/2020).

[33] Statista, *Global recorded music revenue from* 1999 *to* 2019, Web page, Hamburg: Ströer, 2021. [Online]. Available: http://statista.com/statistics/272305/global-revenue-of-the-music-industry (visited on 20/02/2021).

[34] Z. Šikić and Z. Šćekić, *Matematika i muzika*, Croatian, 1st ed., I. Žderić, Ed. Zagreb: Profil, 2013.

[35] A. Vincent. (2019). Lisztomania: the 19th-century pop phenomenon that made Beatlemania look tame. C. Evans, Ed., The Telegraph, [Online]. Available: http://telegraph.co.uk/music/artists/lisztomania-19th-century-pop-phenomenon-made-beatlemania-look (visited on 17/08/2020).

[36] VPRO. (2020). Australia wins artificial intelligence song contest, VPRO, [Online]. Available: http://vprobroadcast.com/titles/ai-songcontest/articles/australia-wins-ai-song-contest.html (visited on 23/12/2020).

[37] ——, *The AI song contest*, Web page, Hilversum: VPRO, 2020. [Online]. Available: http://vprobroadcast.com/titles/ai-songcontest.html (visited on 23/12/2020).

[38] YouTube, *Trending*, Web page, San Bruno: Google, 2020. [Online]. Available: http://youtube.com/feed/trending (visited on 17/08/2020).