Song Popularity Prediction among Countries using Machine Learning

Joan Ficapal Vila[12], **Nicolò Vendramin**[12]

[1]Department of Information Engineering, Politecnico di Milano, Milan

[2]Department of Computer Science, Kungliga Tekniska Högskolan, Stockholm

{joanfv,nicolov}@kth.se

Abstract. *We present Song Popularity Predictor (SPP), a system able to predict the evolution of the top popular song chart in a set of counties starting from the same chart in the previous days.*
The problem belongs to the class of learning and modelling time series and it has been approached employing artificial neural networks with recurrent architectures, in particular long short term memory cells, tuned through multiple series of empirical training and evaluation.
The tool has been evaluated according to the combination of mean average error of the predictions generated on the test set, catching the average distance from the predicted and actual rank of the song in each of the countries, and accuracy on the direction of the trend. The results show an improvement of the score against the baselines used for comparative evaluation.
The core of the model has already been built and tested, nevertheless there is still space to improve the performances by refining the tuning of the parameters, that could not be extremely detailed due to the low computational power on our disposal.Using this model it is possible to forecast the evolution of the charts through time, only by using the history of the same chart and thus it could be a valuable tool to be included in other applications such as a recommendation engine or marketing analysis applied to the music industry.

Keywords*: Forecasting, Music, Neural Networks, Predictive Models, Time Series Analysis.*

Contents

1. Introduction

The music industry is a business that is extremely bounded to the taste of people and, like all similar businesses, suffers from trends and fashions that are temporary. This implies that a song that is popular today may be no longer interesting for the public in a close future and this kind of phenomenon makes it really important for the businesses operating in the music industry to be able to forecast and estimate the evolution of the taste. In order to fullfill this task some tools have been developed to forecast the popularity of songs using different kind of features, but mainly following the content based approach, thus trying to predict the rank or popularity of a song using its characteristic such as the key, the loudness or the name.
The second starting point of this work comes from the fact that is commonly known that some countries share cultural bounds between each other mainly due to historical and geopolitical phenomena. For example it is safe to say that Italy and Spain are "closer" than Italy and China due to their geographic positioning, the history of the two countries, the reduced language barrier and several other factors. This cultural bounds between nations are often obvious and easy to spot, but sometimes rather hidden and surprising, making it difficult to define a map of cultural influences which is clear and well defined.

The following work introduces Song Popularity Predictor (SSP), a model built in the attempt to exploit the phenomenon explained above for the song popularity prediction problem introduced in the first paragraph.

The first contribution is to apply the multivariate time series analysis to the ranking of a song to predict its popularity, and the second is to assess the impact of the cultural bound between countries on the evolution of the chart in the dataset. For the purpose of the latter contribution the authors have decided not to include further features in order to be able to isolate the effectiveness of the use of the ranking in all the countries alone.

The work has been developed and tested basing on the dataset Spotify's Worldwide Daily Song Ranking, publicly available on the Kaggle platform [1].

2. Theoretical Background and Literature Study

Although the problem is a big issue in the music industry, not many publications have been released about possible solutions for it. The most common solution that has been adopted up to now is to apply feature based techniques on several human-readable characteristic recorded in the dataset for each song, such as the key, the name of the song, the artist or the year of release.

Pham et Al. worked on the "Million Song Dataset"[2] to evaluate combinations of classification and regression techniques for song popularity prediction and try to estimate the impact of different features on the effectiveness of the algorithmic solution [9]. Their

[1] Spotify's Worldwide Daily Song Ranking, version of August the 20th 2017.
Available at `https://www.kaggle.com/edumucelli/spotifys-worldwide-daily-song-ranking`

[2] The Million Song Dataset, Columbia University.
Available at `https://labrosa.ee.columbia.edu/millionsong/`

solution approached the problem using a binary classification distinguishing non popular songs from popular ones. A similar approach has been employed by Lin and Newman [6].

The only previous study employing a time series analysis to the problem is the work of Mussmann, Moore and Coventry [7]. They propose a two step model using lyrical content to estimate trajectory labels for the different songs that are used in the following phase to feed a time series to learn forecasts of future popularity. Their work, that is somehow relatable to the one presented in this paper, is still not taking into account any feature related to the different national rankings for the same song and, furthermore, they apply K nearest neighbours as a technique for the analysis of the time series whilst the Song Popularity Predictors exploits recurrent neural networks with LSTM cells.

The report Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market by Salganik, Dodds and Watts explains that social influence accounts as a main factor when it comes to the popularity of a song, somehow giving a base supporting the hypothesis that similarity between countries could play an important role in the estimation of the popularity of a song, assumption that this work aims to leverage and prove [10].

Finally, we would like to recall the work of Skowron et Al. that proved that the inclusion of Cultural and Socio-economic factors in models for genre preferences provides improvement on the precision of the estimations over the baselines, supporting once more the idea that the cultural and socio-economic similarity shifts to musical tastes [12].

3. Research Question and Hypothesis

The question that is at the base of this research is thus to understand whether is possible to employ only the evolution of the ranking of a song in multiple countries as an input for a multivariate time series analysis and obtain a good performing system, proving that the hidden cultural bounds among countries affect the quality of the predictions. Our hypothesis is that using a recurrent neural network with long short term memory cells, we may be able to capture those bounds and to provide accurate predictions of the rankings.

4. Research Methodology

The development of the project has been organised in different phases that will be explained in the following subsections. All the phases have been carried out using on the shelf personal computers (MacBook Air 2015, MacBook Pro 2017) and Python version 3.5. A more detailed list of the external packages used will be given where necessary.

4.1. Data Gathering and Analysis

The dataset has been fetched by the Kaggle repository on its version of the 20th of August 2017. The data is provided in the format of a comma separated version file including line by line the record corresponding to the 200 positions of the 53 + 1 (countries + global) charts for the 230 days between the 1st of January 2017 and the 17th of August of the same year. It contains 4682 different artists and 11932 songs.
Each one of the 2 484 000 lines of the dataset contains the following eight fields:

- Position: the rank in the regional chart.
- Track Name: the title of the track.
- Artist: the interpreter of the track
- Streams: the count of the number of streams in the given regional chart.
- URL: the Spotify resource locator to find the song online.
- Date: the day of the record.
- Region: the code of the country whose chart is considered for the record.

The column Artist, Track Name and URL are not used for the purpose of the analysis cause each sequence was purposely anonymised not to consider authorship or track name in the popularity prediction, to make sure the model only gets intra-countries dependencies and not more trivial ones related to those features, that have already been explored by others in previous works [6].

Between Position and Streams the former has been chosen to base the analysis. As a matter of fact the number of streams, is influenced by the size of the population of the country, the higher or lower penetration of the Spotify application and other heterogeneous effects that would have likely affected the results. The feature Position on the other hand suffers from the opposite problem of flattening differences to a constant step. In fact, it could be that the step of one position hides differences in the number of streams of extremely various size. Nevertheless we considered that the first problem was more relevant and thus, we decided to drop Streams in favour of Position. Date and Region were exclusively used to build the sequences.

For the purpose of this task we employed the external library Pandas[3] to handle the file containing the data.

4.2. Data Preparation

The main objective of the preparation phase was to transform the plain dataset into a collection of sampled sequences to be fed into the recurrent neural network. In order to do so, we employed a simple algorithm that for each required sample extracts a random day, a random song appearing in at least one ranking with a given position in the extracted day and we built the time series starting from that day for n following days. The ”Out of the ranking” position has been encoded with the value minus one and the ranking has been inverted in order to have the ”Out of the ranking” be minor than all the others and to ensure that having $\text{position}_{\text{song1}}$ minor than $\text{position}_{\text{song2}}$ implies that song_1 is ranked worse than song_2.
Each time step is built as a vector of 54 elements representing the ranking in the 53 countries and in the global chart. The target is selected k day after the nth day of the time sequence. To the extracted dataset is applied an hold-out split that keeps only the 80% of the sampled sequences for the training phase, leaving the resting 20% for the test of the system. The selection of k and n is explained in the results section.

4.3. Architectural Study and Implementation

This phase of the project involved the team in the study of the available models for the prediction of time series and the following implementation of the chosen candidate. In

[3]Pandas library, version 0.20.3 for Python 3.5. Documentation at `https://pandas.pydata.org`

the following subsections we go through the study of the methodology narrowing from a broader perspective down to the actual choice that has been made for the implementation of the model.

4.3.1. Theoretical Methodological Framework

Time Series A time series is defined to be "an observation on a stochastic process" [4], or in other terms, a set of points organised in chronological order and expressing the evolution of a certain value, state or figure over a certain time span.
To resume, a time series S of the quantity T is a set of records of T organized in chronological order, i.e: $TS_{\mathrm{T}} = \{s_{\mathrm{t1}} \dots s_{\mathrm{ti}} \dots s_{\mathrm{tn}} \mid \mathrm{s_{tj}}$ follows $\mathrm{s_{ti}}$ implies $\mathrm{t_j} > \mathrm{t_i}\}$.
A time series is defined to be multivariate in case the input channels whose variation is recorded over time are multiple (e.g. rain and humidity levels over a certain period of time).
Different approaches are available to face the time series forecasting problem among which are to be mentioned statistical, probabilistic and analytical approaches such as autoregressive integrated moving average models (ARIMA), diffusion models and the modelling through Markov processes. In recent years machine learning techniques have shown their potential when applied to time series forecasting and in particular Artificial Neural Networks have resulted in being an extremely powerful tool when it comes to the problem of analysing a time series [1].

Recurrent Neural Networks Recurrent Neural Networks are a strict superset of artificial neural networks, disposing of memory cells and where the neurons are connected with direct cycles. They were firstly introduced by Hopfield in 1982 [4], and representative of this class are Elman [2] networks and Jordan Networks [5].
The possibility to include stored information in the computation makes this architectures particularly suitable to be applied on multiple inputs to be interpreted as a sequence. This kind of architectures are generally trained through a technique that is considered an adaptation of standard backpropagation and, because of that is called backpropagation through time [13].
This kind of networks showed empirical evidence to be particularly suitable for time series forecasting thanks to their ability to extend the input output relationship to the whole sequence due to their learning technique. The main limitation of this model is that when the network is unfolded for a big number of steps, the gradient could tend to be amplified or suppressed causing respectively the exploding or vanish gradient phenomena.

Long Short Term Memory LSTM (Long short term memory) is a recurrent architecture that "is designed to overcome these error back-flow problems"[5]. As a matter of fact, this model solves the problems of RNN and other proposed solutions applied to learn time structures "by enforcing constant error flows through constant error carousels within special units called cells"[6]. An LSTM memory "contains a node with a self-connected recur-

[4] Pages 1-3 of the book referenced at [8]
[5] Introduction (Section 1) in [11]
[6] Abstract of the document [3]

rent edge of weight 1, ensuring that the gradient can pass across many time steps without vanishing or exploding"[7]. Empirical evaluation has shown that LSTM outperform traditional Recurrent Neural Networks in terms of capability to learn long term dependencies.

4.3.2. Designed Architecture and Framework Definition

For the scope of the project we focused the attention on recurrent architectures for time series and in particular, given the lack of extensive processing capability, the authors decided to try out only one single architectural model, leaving to possible future work the option to test the performances of different architectural choices.

Given the theoretical framework provided at 4.3.1. we decide to implement a simple architecture with a single layer of long short term memory cells, stacked on the input layer and followed by a dense output layer with one output neuron for each one of the 54 charts that the networks tries to forecast. The described architecture is summarized in Figure 1 by an explanatory image in which the network is sketched in a rather simple way, to help the reader to easily understand the structure of the model.

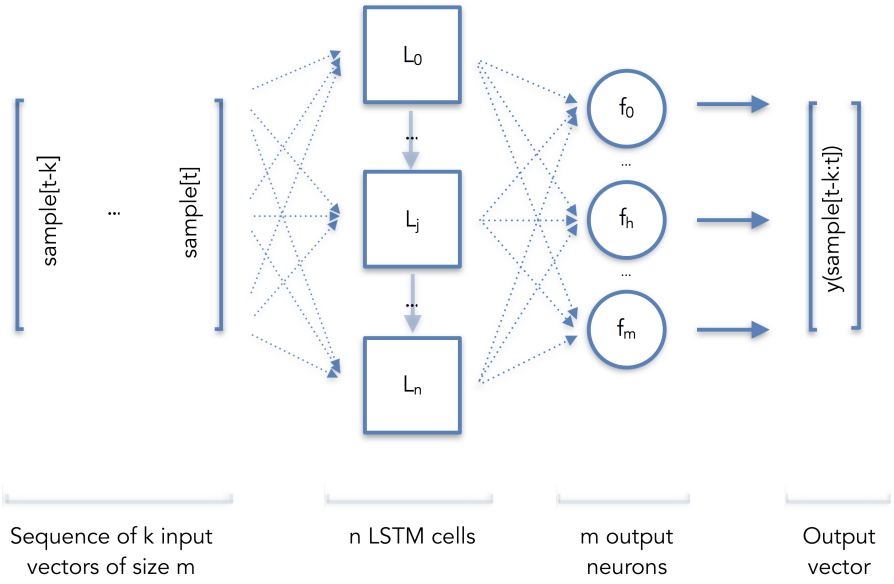


Figure 1. Schematic representation of the target architecture.

The implementation of the model was carried out using the Keras [8] library for deep learning in Python, running over Tensorflow [9] backend.

[7] Page 16, section 3.3.1 in [14]

[8] Keras library, version 2.1.1 for Python 3.5. Documentation at `https://keras.io`

[9] Tensorflow library, version 1.4.0 for Python 3.5. Documentation at `https://www.tensorflow.org`

4.4. Metrics and Baseline Definition

After a study of the possible metrics to be applied for the evaluation of a time series analysis, *mean average error* has been chosen both as the metric to be minimized by the learning process and as proxy to evaluate how well the system is performing. *Mean average error* compares the predicted output vectors with the real output of the sequence and averages for each prediction the absolute value of the difference between the rank prediction and the real value for each country. The figure is successively averaged over the whole set of predicted values.

This metric directly shows how far the predicted rank is in average from the target value to be estimated. Although it's clarity makes it for sure a good evaluation metric, we decided to combine it with a second one used to be able to understand also the quality of the direction of the prediction, together with the one of its module. In order to do so we define for each predicted vector and target values vector respectively two binary vectors. Class 0 means that the rank decreases with respect to the last element of the sequence, whilst class 1 represent that the rank increases or remains the same. Using those newly defined vectors we can apply an *accuracy* metric over all the prediction to understand how often the predicted and real direction of the evolution agree. *Accuracy* is only evaluated for those input points whose final ranking is not close to the boundaries [-1,+200], avoiding to over-boost the score thanks to trivial cases (e.g.: a song not appearing in one of the charts in the considered sample).

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|e_t|\,, \qquad \text{Accuracy} = \frac{t_\text{p}}{t_\text{p}+t_\text{n}}$$

with e_t ={position wise difference between prediction and target value},

n ={number of samples}, t_p ={true positives}, t_n ={true negatives}.

Since no previous comparable study has been done it was not possible for us to reimplement and use an existing solution as a baseline to test the effectiveness of our approach and thus, we decided to realise some customised and intuitive baselines that seem reasonable to evaluate the quality of the predictions of SPP. Below we briefly describe the five baselines that have been used:

- **Random vector baseline (RV):** this is the typical bottom baseline that simply estimates the value to be predicted with a vector randomly initialized in each position with a value extracted between -1 and +200 with uniform distribution.
- **Constant baseline (C):** intuitively is safe to think that while forecasting the evolution of a ranking the difference in the position of a song in the ranking in a time span of days may not vary considerably. Thus, the second designed baseline simply predicts the same ranking of the last day of the sequence.
- **Random increase baseline:** basing on the same assumption as above, the third baseline simply predicts the ranking of the last day of the sequence altered on each position by a factor extracted from a Gaussian with mean 2 and standard deviation 1 that is with uniform probability either added or subtracted to the previous value.

- **Average baseline (AVG):** this baseline estimates the target as the average for each position of the rank over the days considered during the sequence.
- **Linear regression baseline (LR):** this baseline applies standard linear regression to the flattened input. Instead of considering the sequence of days, the ranks on each day of the sequence are stacked on a single input array that receives as target the day to be predicted. This baseline tries to express the output as a linear combination of the values in the previous days.

4.5. Hyper-parameter Selection and Model Tuning

The architecture introduced in the previous section has been trained and validated with different settings of number of cells, optimizer, activation function of the recurrent layer , activation function of the dense layer and dropout percentage before the output layer running a grid search with the configuration reported below, applied on a five fold cross validation split of the training data.

Parameter	Values				
Optimiser	Adam	Rmsprop	Sgd		
Recurrent Activation	Sigmoid	Hard Sigmoid	Linear		
Dense Activation	Linear	Adjusted Linear**	Custom Sigmoid*		
Number of Cells	100	200	400	600	800
Dropout Probability	0.1	0.2	0.3	0.4	0.5

**: f(n)= $\begin{cases} -1, & \text{if } n < -1 \\ 200, & \text{if } n > 200 \\ n, & \text{otherwise} \end{cases}$ *: f(n)= (sigmoid(n) * 201) - 1

Figure 2. Values used during the grid search of the best hyper-parameter setting.

4.6. Testing and Evaluation of the Results

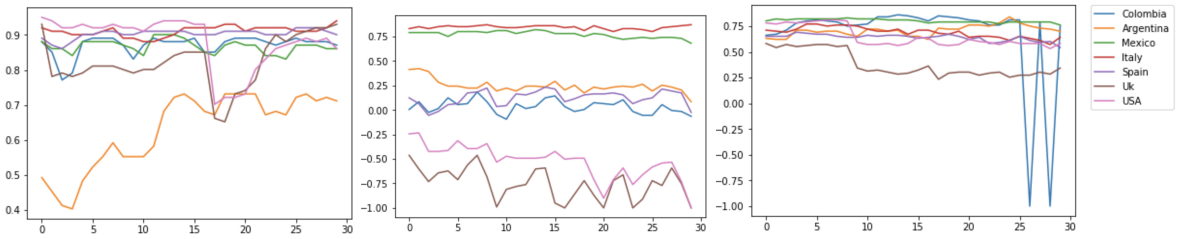
The main focus of this phase was to design and run a set of meaningful tests to verify the operation of the tool under different working condition. In particular, the design of tests was carried out in order to assess the variation in the performances of SPP as a response to the modification of the input shape i.e. the length of the sequences and the distance between the last day of the sequence and the value to predict.

5. Results and Analysis

In this section we report the empirical results obtained in the different phases illustrated in the previous section. Due to the limited computational power on our disposal the evaluation is limited both in the number of attempts and in the size of the data extracted and used for the training of the model.
Starting from the plain data we extracted a multitude of differently shaped data sets whose characteristics are introduced later in Figure5a.

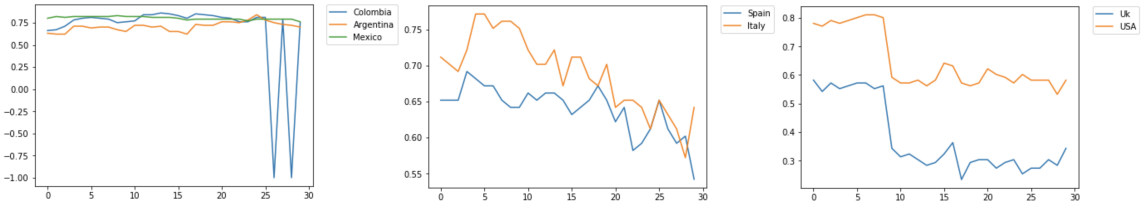
5.1. Quantitative Analysis

For the purpose of this task we extracted sequences of thirty days for songs with three different level of popularity, more specifically top popular songs, medium popularity songs and low popularity songs (respectively appearing in the charts at least with rank 15, 50 and 100 in the starting day of the sequence). We performed an empirical observation of meaningful sample sequences to understand whether the hypothesis of this research was also mirrored as a macroscopic effect, already identifiable through the visualization of the evolution of the sequences. In order for the visualization to be human readable we randomly selected seven countries between the 53 for which we have data.
The results shown in Figure3 illustrate that there seem to be correlated evolution of charts that tend to be stronger as the popularity of the song grows. The intuitive interpretation is that the more a song is ranked and the more is likely to spread to several countries, causing similar evolution in countries that are somehow related.



(a) Low Popularity (b) Medium Popularity (c) High Popularity

Figure 3. The evolution over 30 days for 7 randomly selected countries of three songs with different levels of popularity.



(a) Colombia, Argentina and Mexico (b) Spain and Italy (c) UK and USA

Figure 4. The chart of Figure3c in which the countries are grouped in clusters of intuitively related countries.

The results of this phase showed that the rank over time of songs in a country does not evolve independently, but on the contrary the evolution groups countries in clusters that show similar trends. Those clusters change from song to song implying that the relationship behind this phenomenon are not trivial and clear connections that can be defined once for all.
In Figure4 we see the trends of each cluster splitting the plot shown before in Figure3c in three different subplots containing the countries divided according to the similarity of the evolution of their ranking.
The value on the x and y axis of the plots in Figure3 AND Figure4 are respectively the

rank of the considered song (mapped from [-1:200] to [-1:+1]) and the day of the sequence.

5.2. Model Selection, Training and Performance Evaluation

The grid search cross validation with the parameters specified in the table reported in Figure2 signaled as best performing model the following configuration: Adam optimiser, customised sigmoid activation in the dense layer, hyperbolic tangent as activation for the eight hundred LSTM cells of the recurrent layer and with a dropout probability of 20%. The model has been trained and evaluated over a set of 11 different datasets with different characteristics in order to be able to understand how the size of the dataset and the distance of the target day from the sequence affect the performances of the model. The different configurations are reported the table shown in Figure5a, and the result obtained by the model and the baselines are reported in the tables of Figure5b and Figure5c, respectively showing *accuracy* and *MAE* of the different baselines and the model on the test set split of the specified dataset. The parameter n and k referenced in the section 4.2 are here respectively indicated as "Length of the Input Sequence and Distance of the Target from the Last Input Day".

DatasetID	Number of Samples	Length of the Input Sequence	Distance of the Target from the Last Input Day
D_A	1500	7	1
D_B	1500	7	7
D_C	1500	7	15
D_D	1500	7	21
D_E	5000	7	1
D_F	5000	7	7
D_G	5000	7	15
D_H	5000	7	21
D_I	10000	7	1
D_J	10000	7	7
D_K	10000	7	15

(a) Description of the different dataset configurations

DatasetID	acc_{lr}	acc_c	acc_{ri}	acc_{avg}	acc_{spp}
D_A	0.78	0.28	0.50	0.52	0.88
D_B	0.71	0.29	0.50	0.52	0.81
D_C	0.72	0.26	0.50	0.48	0.79
D_D	0.74	0.23	0.48	0.50	0.80
D_E	0.79	0.33	0.50	0.52	0.86
D_F	0.77	0.30	0.49	0.52	0.85
D_G	0.75	0.29	0.50	0.50	0.82
D_H	0.75	0.26	0.51	0.50	0.82
D_I	0.82	0.30	0.49	0.52	0.88
D_J	0.79	0.27	0.50	0.52	0.86
D_K	0.77	0.27	0.50	0.49	0.84

(b) Test *Accuracy*

DatasetID	mae_{lr}	mae_c	mae_{ri}	mae_{avg}	mae_{spp}
D_A	15.25	14.95	15.79	15.23	6.02
D_B	14.92	12.59	13.45	12.30	6.59
D_C	13.83	9.99	10.88	10.15	5.77
D_D	14.34	12.30	13.18	12.09	6.66
D_E	11.39	13.06	13.91	12.84	5.39
D_F	9.91	10.65	11.54	10.69	4.53
D_G	11.25	12.70	13.56	12.76	6.08
D_H	10.90	12.11	12.97	12.13	5.65
D_I	10.04	13.19	14.05	13.00	4.83
D_J	9.65	12.31	13.18	12.19	4.85
D_K	9.82	12.26	13.13	12.16	5.27

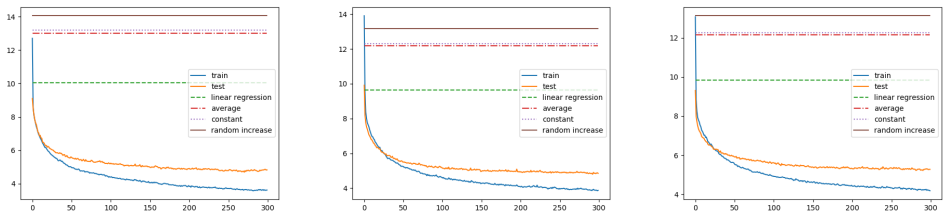
(c) Test *MAE*

Figure 5. Description of the characteristics of the 11 used dataset (a) and related *Accuracy* (b) and *MAE* (c) scores of the model and baselines.

The empirical results show that SPP is able to outperform all the given baselines in its forecasting ability even with a raw tuning and a limited training. As a matter of fact, in all the experiments that we run SPP results in better *accuracy* and lower *MAE* than all the other models. For what the *accuracy* is concerned all the baselines exception made

for the linear regression baseline show performances comparable or lower to the ones of a random guesser, and even in the case of linear regression the results of the model presented in this paper are still at least 0.06 points better, meaning that the SPP is able to understand and forecast the increasing/decreasing trend for a song with at least six percentual points of advantage with respect to his better competitor among the baselines. For what the *MAE* score is concerned, all the baselines including linear regression show poor performances when compared to the predictions of Song Popularity Predictor. In fact, the mean absolute error of the trained model is low even when the trained takes place on a small amount of data and is generally around half of the score obtained by the most performing baseline. We also notice that depending on the single experiment the best performing baseline changes while the dominance of SPP both in terms of accuracy and mean average error remains solid. These observations prove that SPP shows a consistent capability to forecast both the rank and the direction of the trend with an high degree of precision.

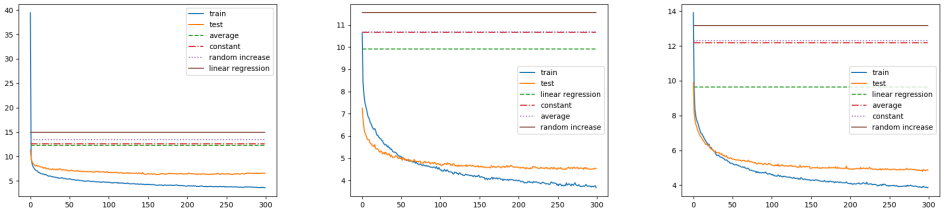
The experiments also show some properties with respect to the dependency of the performances of the system from the distance of the target chart to be forecast from the last day of the input sequence and the size of the dataset. In terms of this analysis is interesting to notice that the performances of the model keep good even when it tries to forecast the ranking of two or three weeks later, as a matter of fact we do not notice a considerable drop in the *MAE* and *accuracy* figures in such conditions. As expected the model reaches better performances increasing the size of the data both in terms of accuracy and mean absolute error. Furthermore the empyrical evaluations shows that increasing the amount of available data for the training the performances of the model vary less on the variation of the distance of the target.



(a) MAE on D_i (b) MAE on D_j (c) MAE on D_k

Figure 6. Plots reporting the MAE of the model and baselines with increasing distance of the target day from the input sequence. The subplot caption references the dataset used for the image using the DatasetID of Figure5.

Finally we notice from the comparison of train and test loss evolution that the model tends to overfit, signaling that a finer tuning of the dropout or the adoption of other generalization techniques is required. At the same time the model does not completely reaches zero error on training data leaving open the hypothesis that probably a more complex model could improve the fitting of the input sequences.



(a) MAE on D_b (b) MAE on D_f (c) MAE on D_j

Figure 7. Plots reporting the MAE of the model and baselines with increasing size of the dataset. The subplot caption references the dataset used for the image using the DatasetID of Figure5.

6. Discussion

The previous paragraphs illustrate how Song Popularity Predictor is able to predict with high degree of precision the evolution of national rankings by putting together the time series of multiple countries. This shows that the hypothesis formulated in Section3 is confirmed after the empirical evaluation of the system. As a matter of fact, SPP shows that only by considering the chronological evolution of the ranking, along with the same in other countries, we are able to build a predictive model that leverages on those cultural and socio-economic factors, and captures how they are reflected in the media markets such as the music industry. This result is aligned with what stated by Skowron et Al. [12] and supports the idea that considering the multinational dimensionality of such problems improved forecasting models can be achieved.

This work leaves many open questions that could be interesting to be explored in future work. At first, it would be appreciated to verify how much the performances of the system can be pushed by applying a finer and more extensive training. It would also be relevant to explore different architectures within and outside the field of Recurrent Neural Network to compare the performances of such systems on the evaluation of the multivariate time series problem. From a different perspective, this work could be included in further research by trying, for example, to plug SPP in a more complex predicting system including feature based modeling to evaluate how much the multinational dimensionality improves the predictions of the system. The work could also be integrated in recommendation tools in order to try to improve such metrics as novelty and diversity by recommending songs that are likely to become popular in the future.

Although there are no direct sustainability issues related to our research, we took care to cover this aspect and to make sure that all possible ethical and sustainability implications were considered. We responsively took care not to harm or offend anyone in the selection of our topic, methodology and without including any bias or pre-defined orientation that could be considered as discrimination. To respect the principle of trustworthiness, the implementation of the work is released and made available under GNU General Public License, at the URL `https://github.com/nicolovendramin/Song-Popularity-Predictor` on Github. In this way, anyone will be able to read it, evaluate it, and run it in order to verify

the reliability of our assertions, contribute to our research or use our product as a starting point for future work, relying on collaborative values for further expansion of the work.

References

[1] O. Claveria and S. Torra. Forecasting business surveys indicators: neural networks vs. time series models. *AQR Working Papers*, 12:1–2, 2013.

[2] J. L. Elman. Finding structure in time. *Cognitive Science, A Multidisciplinary Journal*, 14:179–211, 1990.

[3] F. Gers. Long short-term memory in recurrent neural networks. *Unpublished Phd Dissertation, Ecole Polytechnique Federale de Lausanne*, 2001.

[4] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceeding of the National Academy of Science*, 79:2554–2558, 1982.

[5] M. I. Jordan. Serial order: A parallel distributed processing approach. *Advances in psycology*, 121:471–495, 1986.

[6] K. Lin and R. Newman. Predicting song popularity. *Unpublished paper, Northwestern Univeristy*, 2017.

[7] S. Mussmann, J. Moore, and B. Coventry. Using machine learning principles to understand song popularity. *Unpublished Paper, Stanford University*, 2014.

[8] E. Parzen. A survey of time series analysis. *Tecnical Report, Applied Mathematic and Statistic Labs, Stanford University*, pages 1–3, 1960.

[9] J. Pham, E. Kyauk, and E. Park. Predicting song popularity. 2015.

[10] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311:854–856, 2006.

[11] J. Schimidhuber and S. Hochreiter. Long - short term memory. *Neural Computation*, 9:1735–1780, 1997.

[12] M. Skowron, F. Lemmerich, B. Ferwerda, and M. Schedl. Predicting genre preferences from cultural and socio-economic factors for music retrieval. *European Conference on Information Retrivial*, pages 561–567, 2017.

[13] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1990.

[14] J. B. Zachary Chase Lipton and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv:1506.00019*, 2015.