INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Machine Learning

Report
Predicting Spotify Song Popularity

Benzitouni Fethi
Bertin Alexandre
Hoang Van Hao
Nguyen Hai Vy

4 GMM A Promo 55

- 19 mai 2021 -

Sommaire

Introduction

Spotify is the most widely used music streaming service nowadays. This application currently catalogs about 60 million tracks. To choose the songs to make available or to recommend certain songs to its users, it is interesting to be able to predict the popularity of a song. This is what we will try to do here.

We have 10 000 songs each characterized by 15 variables : the positiveness of the track, the release year of track, the relative metric of the track being acoustic, the relative measurement of the track being danceable, the length of the track in milliseconds, the energy of the track, the relative ratio of the track being instrumental, the primary key of the track, Bb meaning A] or B [, the relative duration of the track sounding like a live performance, the relative loudness of the track in the typical range [-60,0] in decibel, a binary value representing whether the track starts with a major [encoded 1] chord progression or not [encoded 0], the relative length of the track containing any kind of human voice, the tempo of the track in Beat Per Minute, the popularity of the song rated by a number between 0 and 100 and another variable pop.class correspond to the same thing but in this case, the popularity is rated by a letter from A to D.

The idea is to apply various machine learning methods seen in class on these data to draw a prediction of the popularity by going through a supervised regression problem and then through a supervised classification problem. We will first perform a descriptive analysis of the data. Then we will implement linear machine learning methods. Finally, we will implement more sophisticated methods such as CART, random forest, or neural network.

I Preliminary exploration of the data

In this section, we are going to apply some basic statistical methods to better know our dataset. We are first going to do some descriptive statistics and then we will see what the principal component analysis method gives us for this database.

I.1 Unidimensional Descriptive Statistics

First of all, we plot individually the boxplots of all quantitative data as shown in Figure 1. What gets our attention is the shapes of our boxplots. Some are flattened while others are extended. This is explained by Figure 2.

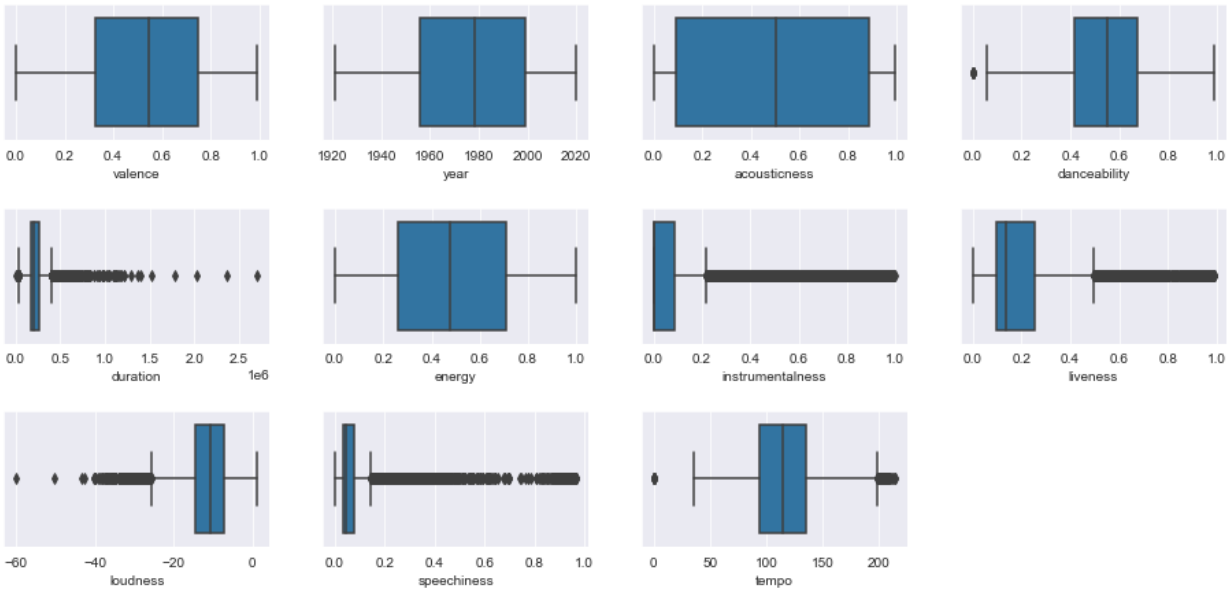


FIGURE 1 – Boxplots of 11 quantitative variables

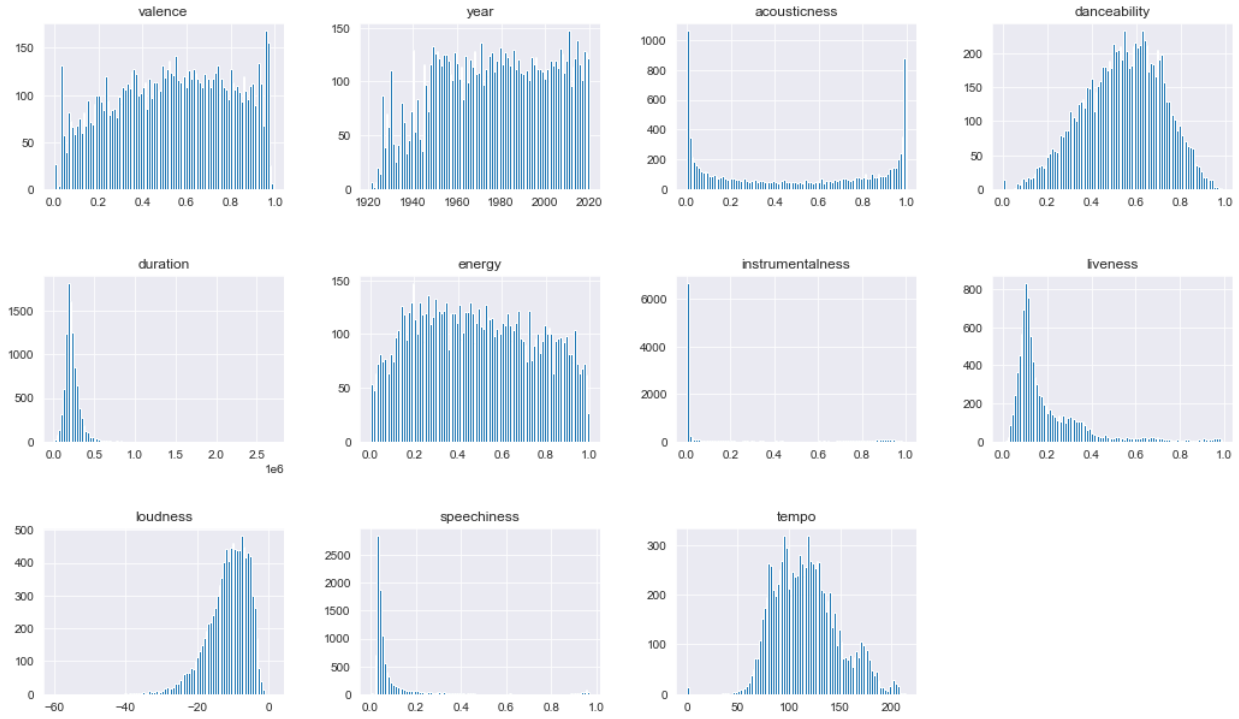


FIGURE 2 – Histograms of 11 quantitative variables

We can see that some distributions are homogeneous and some distributions represent a gaussian distribution as example danceability. While some others distributions can be represented by a distribution following a poisson distribution as example speechiness. The variable acousticness seems to follow a uniform distribution.

Regarding Figure 1 and Figure 2, we can see a link between the shape of the boxplots and the hypothetical distribution. For, a centered boxplot where the 1st and the 3rd quantil are relatively close we link to that as a gaussian distribution. For, a boxplot centered on the extreme left of the display and where we can see a lot of outliers, we link to that as a poisson distribution. We can conclude a large variety of a hypothetical well known distribution are followed by our variables.

I.2 Multidimensional Descriptive Analysis

Here we are interested in knowing more about the plausible links between our different variables. On Figure 3 we can see all boxplots flattened apart from duration, which means that the variance of the variables is not on the same scale. So we need to normalize the variables for better analysis. The boxplots after normalizing are shown in Figure 4.

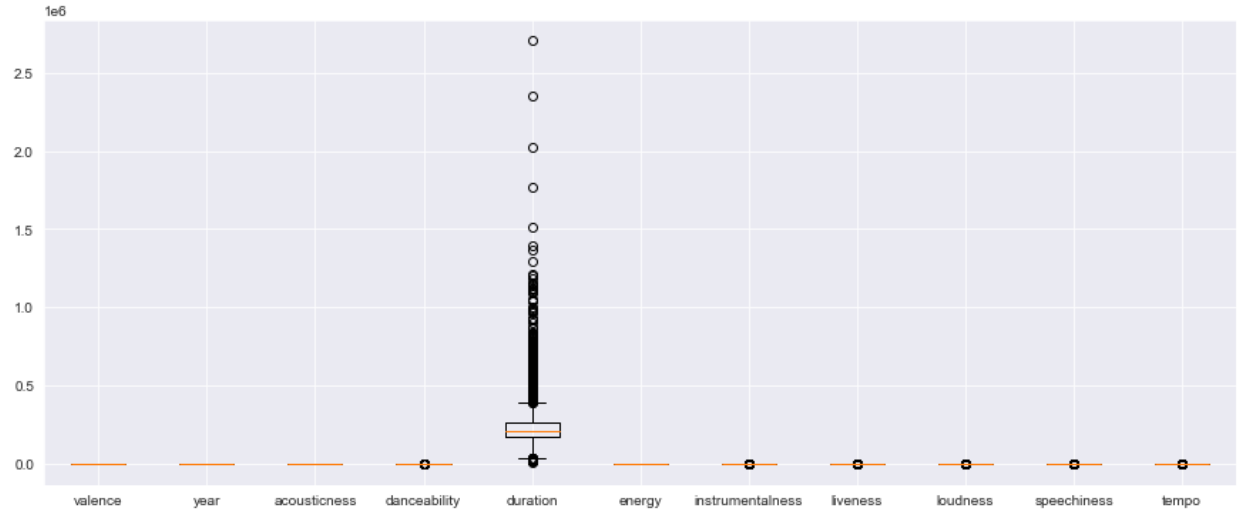


FIGURE 3 – Boxplots of 11 quantitative variables without normalizing

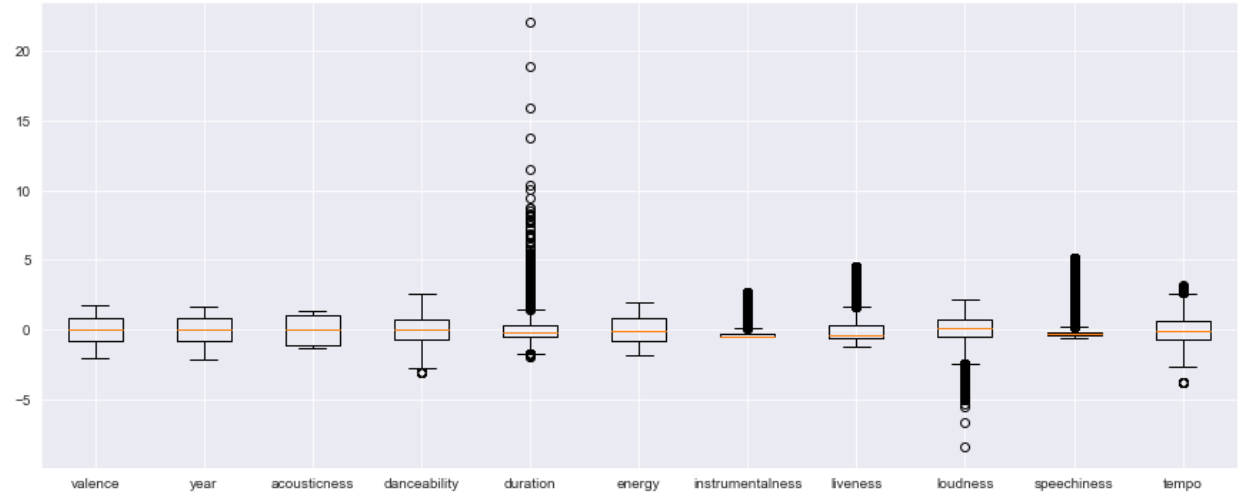


FIGURE 4 – Boxplots of 11 quantitative variables with normalizing

From Figure 4 we can see the asymmetry of 5 variables (duration, instrucmentalness, liveness, loudness and speechiness). When reviewing a boxplot of "duration" and loudness", we observe a lot of outliers that are located outside the whiskers of the boxplots. Then, we will attempt to find out links between quantitative variables. The figure 5 is the correlation matrix of all quantitative variables. Popularity is negatively correlated to acousticness(-0.575481) while it's positively correlated to year(0.858713), energy(0.496238), loudness(0.463429). Acousticness is negatively correlated to popularity(-0.575481), energy(-0.748026), loudness(-0.558335), year(-0.616553). Energy and loudness are strongly correlated (0.780850).

With this matrix of correlation we can easily conclude that a "good" music is recent, loud, energetic and a low acousticness score.

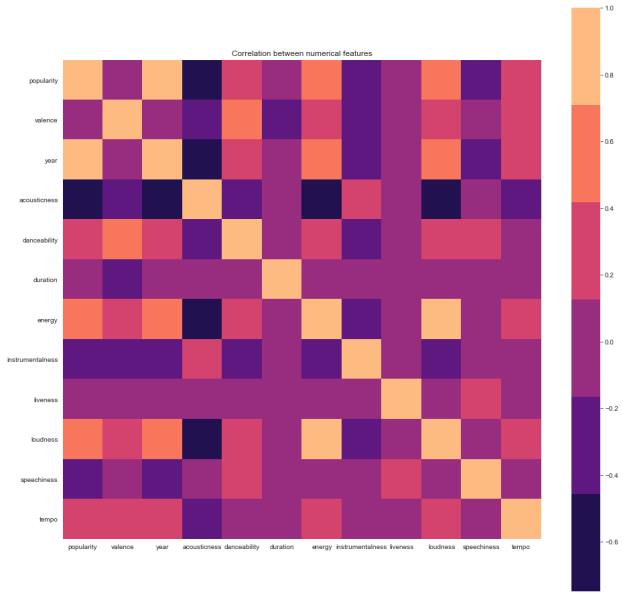


FIGURE 5 – Matrices correlation of 12 quantitative variables

The scatter graphs in Figure 6 is a better graph to show us the linear relation or not between variables. For example, we can observe a linear relation between popularity and the years.

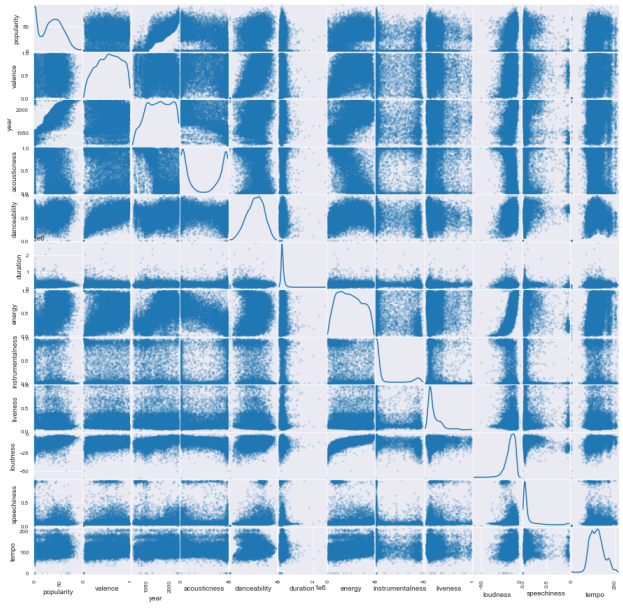


FIGURE 6 – Scatter graphs of 12 quantitative variables

We can say that :

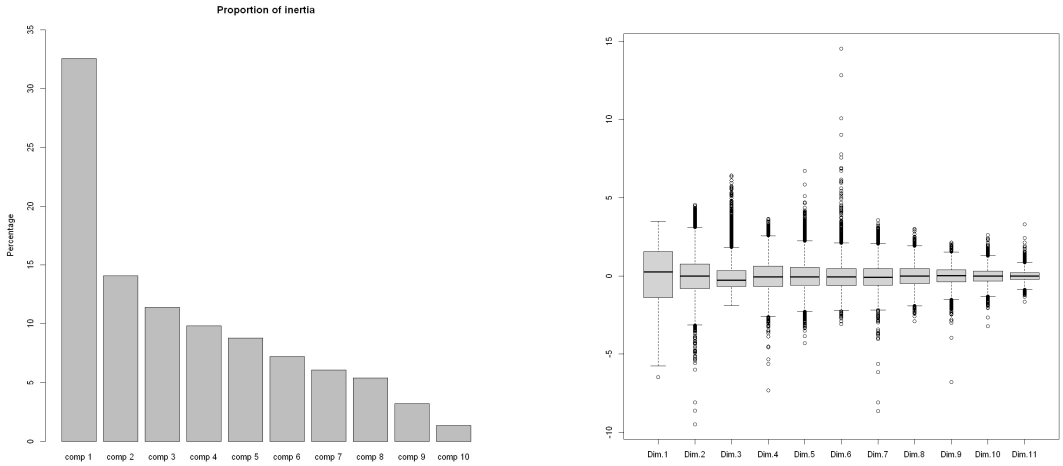
— The low acousticness score of recent musics can be explained by the use of technology. For example, the use of news devices in musics such as electric guitars or synthesizers.

— Loud and energetic are two variables which are naturally linked. The scientific term of loudness is sound intensity. The sound intensity is linked to an energy per unit area (W/m^2). Physically, the loudness is strongly linked to an energy.

I.3 Principal Component Analysis

We currently have 12 quantitatives explanatory variables in our datasets, after removing some variables as explained above. What we will try to do here is to reduce the dimension of our problem

by trying to explain the maximum inertia explained by our 25 variables in a lower dimension. For this purpose, there are different methods of variable reduction in the scientific literature. The most generalised, used and simple method is principal component analysis. The PCA is a process which aims to change the coordinate space. All news variables are linearly uncorrelated. The process to construct these news variables allows for the user to choose those which explains the most the inertia of the system studied. There are 12 quantitative variables, thanks to PCA we reduce them to new variables. Then we will try to interpret them.



(a) Percentage of variance explained by the first 10 components

(b) Boxplots of first 11 principal components

FIGURE 7 – PCA graph 1

According to Figure 7a, we observe that more than eighty percents of variance is explained by the first 6 components. We can have the same conclusion using figure 7b by observing that from the 7^{th} component, the explained variance is insignificant.

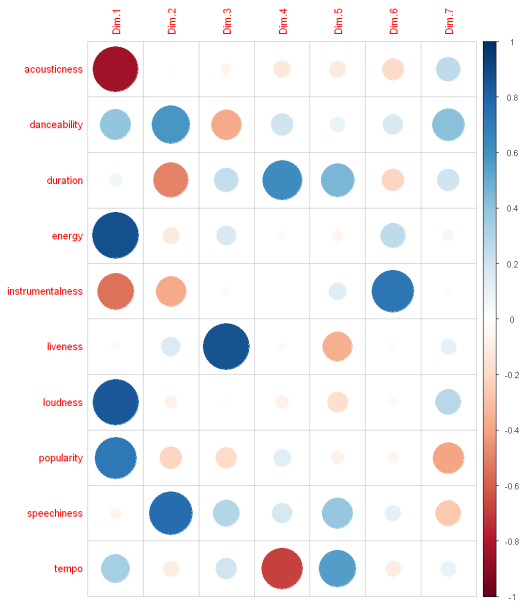
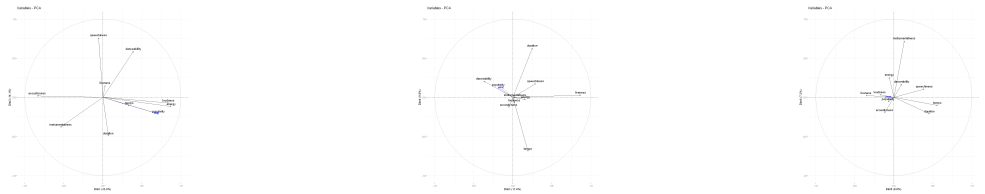


FIGURE 8 – Corrplot quantitatives variables and news variables



(a) Dim 1 vs Dim 2 (b) Dim 3 vs Dim 4 (c) Dim 5 vs Dim 6

FIGURE 9 – Variables factor map

Figure 8 shows us the covariance matrix between the components considered built thanks to PCA and the 12 quantitative variables. As we observed previously, the first meta variable (dim1) is positively explained by the popularity, loudness, energy, and a bit explained by the danceability and the tempo while it's negatively explained by the acousticness and a bit explained by the instrulmentaness. The first meta variable describes well what we saw previously. On the corrplot, the variable year is missing. However it's shown on Figure 9. Figure 9 shows that the variable year and the variable popularity are strongly linked.

I.4 Conclusion

According to the unidimensional and multidimensional descriptive analysis, we can suppose that the popularity is directly strongly linked to year,loudness,energy and acousticness. In this part, we didn't use the categorical variables in our hypothesis. That's why our final is : the popularity directly strongly linked to year, loudness, energy, and acousticness, lightly linked to the other quantitative variables and can be linked to keynote and mode. If this part has allowed us to intuit the most important variables for the prediction of the popularity of music, we will now implement some machine learning methods that will allow us to affirm or deny our hypotheses

II Comparing the performance of some models

In this section, we are going to implement all machine learning methods studied in class for our problem. We will consider the regression problem and the classification problem at the same time to compare the performance of each method.

II.1 Data preparation : Train-Test Data splitting and Normalization

The data-set is composed of $O = N + M$ with O the totality of songs. We will split into two data sets. N songs are randomly selected for forming a data for training while the others M songs are formed in a data-set used for testing our models. In our example $card(N) = 8000$ songs and $card(M) = 2000$ songs.

This split is essential to compare and choose the most suitable model for our data. The learning data will allow us to build the model and the test data will allow us to evaluate the capacity of prediction of the built model.
However, for some methods we have to tune the parameters. Cross-validation (K-folds) allows us to optimize the parameters.
A good method for avoiding the overfitting is the bagging. From the dataset N, we can create m sub training set noted N_i with $\forall i \in [1 : m]$. For each N_i is a sub training set by uniform sampling with replacement from N.
From the previous section, we realize that data normalization is very important. So, we will perform all methods with the normalized data set.

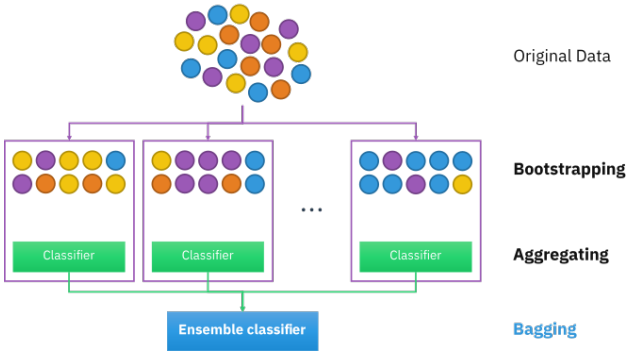


FIGURE 10 – An illustration of Bagging

II.2 Linear regression

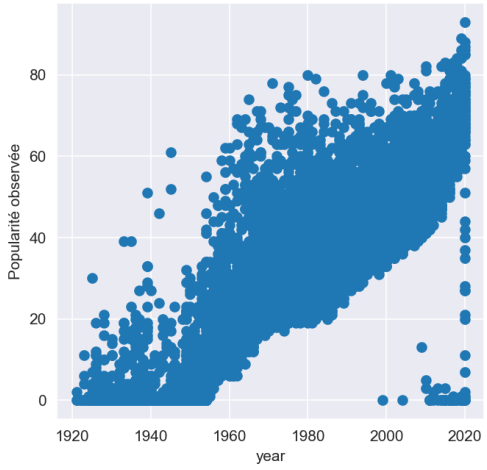
II.2.1 Linear Regression without penalisation

In this section, we assume that the prediction of the popularity Y could be written as a linear combination of explanatory variables plus random noise noted ϵ. Hence we have $Y = X\beta + \epsilon$ where X is the matrix of the observed explanatory variables, β is the vector of the coefficients of the linear combination and ϵ is the vector of the random noises. Therefore our problem is to solve the following optimization problem :

$$\min_{\beta} ||Y - X\beta||^2$$

So, the aim is to obtain the least squares estimator. The preliminary problem that we have to solve before linear regression is to find the property of variable *year* as we do not know if *year* should be used as qualitative or quantitative variable. To answer this question, we plot the plot

popularity versus *year* as shown in Figure 11a. By observing this plot, we see that there is a strong correlation between *year* and *popularity*. So we can use *year* as quantitative variable in the following part.



(a) Scatter graphs of Popularity dependent on Year

	A	B	C	D
A	73	19	1	3
B	145	443	136	5
C	18	130	371	55
D	2	6	50	543

(b) Confusion matrix of linear regression

FIGURE 11 – Scatter graphs and Confusion matrix

Now, we can delve into the linear regression problem. Figure 11b shows the confusion matrix obtained by implementing simple linear regression in the test data. The error of prediction is equal to $28,5\%$. One can say that the prediction is not very good in this case. Since linear regression is a basic method, which gives very good results in the case of linear problems, it is logical that in our case we do not get very good results with this method.

More sophisticated methods of linear regression have emerged : these are known as penalty methods. Indeed in our case here we do not impose anything on our parameter which can lead us to a complex model. The idea of penalisation is to have a less complex model avoiding over-fitting and therefore to obtain a more generalised model.

II.2.2 Linear Regression with Lasso penalisation

A first method of penalisation is the lasso penalty. In this one we add to our least squares problem the norm 1 of our parameter β multiplied by a regularisation parameter λ. Hence our problem become :

$$\min_{\beta} ||Y - X\beta||^2 + \lambda||\beta||_1$$

By doing that we want to find the sparsest solution. Here are the results obtained by applying this method.

	A	B	C	D
A	48	12	0	2
B	118	415	135	10
C	23	137	392	70
D	1	8	50	579

FIGURE 12 – Confusion matrix of linear regression with lasso penality

The error of prediction equals to $28,45\%$. Optimal λ equals to 0.1028271 obtained by cross validation.

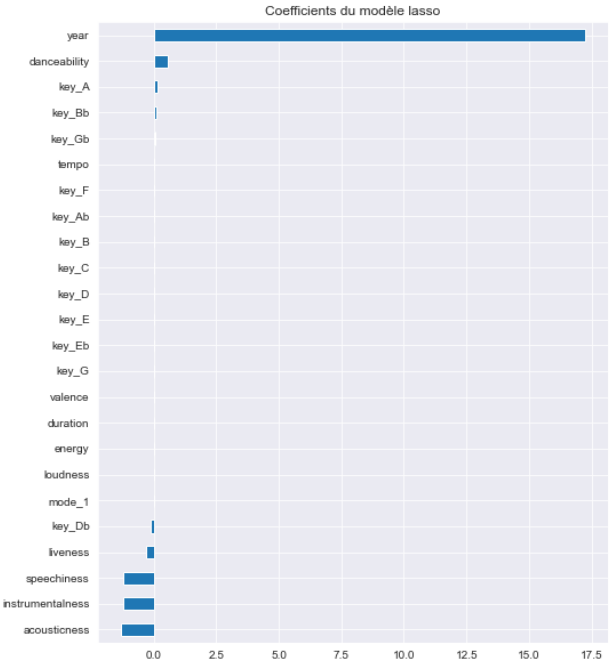


FIGURE 13 – The importance of each variable of the Linear Regression model with Lasso penality

Lasso regression method gave an output of 8 different features with nonzero coefficient estimates. The Lasso penalisation is slightly better to predict notes. Plus the lasso method allows us to show how the variable year is important compared to other variables. As we said before, year is highly correlated to the popularity. The lasso method also shows that acousticness and year are with opposite signs.

II.2.3 Linear Regression with Ridge penalisation

A second penalization is the Ridge penalization. Here instead of a norm 1 we use the norm 2 and the problem to resolve is :

$$\min_{\beta} ||Y - X\beta||^2 + \lambda||\beta||^2$$

	A	B	C	D
A	33	4	1	1
B	185	450	132	7
C	18	139	381	72
D	2	5	44	526

FIGURE 14 – Confusion matrix of linear regression with Ridge penality

Optimal λ equals to 1.882495 obtained by cross validation. The penalisation by Ridge is not effective to predict the popularity of songs. We record an error of prediction above 30%. The error of prediction equals to $30,5\%$.

Finally, it seems that the best method among the linear regression is the ridge penality coupled with the λ_{min}. Plus, the ridge is a better m

All confusion matrix highlight that it is difficult to classify well the songs with high popularity. For most of the music rated A, they are misclassified. We are now going to apply linear method but in a classification problem.

II.3 Logistic Regression

In our case, we will not use a binomial distribution because the songs are noted by four letters. Thus, we have to use a multinomial distribution. We used the package glmnet. Glmnet solves the problem :

$$\min_{\beta_0,\beta} \frac{1}{card(N)} \sum_{i=1}^{card(N)} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1]$$

Where l is the negative log-likelihood of a multinomial distribution. α is the elastic net penalty. If $\alpha = 0$ then it's a ridge regression and if $\alpha = 1$ then it's a lasso regression. λ controls the penalty if $\lambda = 0$ then it's a problem of regression without penalization (we test the full model). In the logisitic regression, we have to specify a linear predictor, a link function and a random component. These items are contained in the object class in GLM. As we said previously, we have to tune parameters thus we will use cv.glmnet as a function.

II.3.1 Logistic Regression without penalisation

The problem is :

$$\min_{\beta_0,\beta} \frac{1}{card(N)} \sum_{i=1}^{card(N)} w_i l(y_i, \beta_0 + \beta^T x_i)$$

	A	B	C	D
A	43	127	18	3
B	11	405	139	6
C	1	124	411	50
D	4	13	31	591

TABLE 1 – Confusion matrix of logistic regression without penalisation

The error of prediction equals to $0,275\%$.

II.3.2 Optimisation of model by penalisation Lasso

The problem is :

$$\min_{\beta_0,\beta} \frac{1}{card(N)} \sum_{i=1}^{card(N)} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda||\beta||_1$$

	A	B	C	D
A	43	126	19	3
B	10	407	138	6
C	1	122	413	50
D	4	13	56	589

TABLE 2 – Confusion matrix of logistic regression using Lasso

The error of prediction equals to $0,274\%$. Optimal λ equals to 0.0002257419 obtained by cross validation.

II.3.3 Optimisation of model by penalisation Ridge

The problem is :

$$\min_{\beta_0,\beta} \frac{1}{card(N)} \sum_{i=1}^{card(N)} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda ||\beta||_2^2$$

The Ridge penalisation is out of the competition. The error of prediction is more than 30%

	A	B	C	D
A	3	164	19	5
B	0	410	138	13
C	0	131	368	87
D	0	17	54	591

TABLE 3 – Confusion matrix of logistic regression using Ridge

The error of prediction equals to $0,314\%$. 10-fold cross-validation is used to choose the optimal value of λ which turns out to be equals to 0.03511813.

II.3.4 Optimisation of model by an elastic penalisation

The problem is :

$$\min_{\beta_0,\beta} \frac{1}{card(N)} \sum_{i=1}^{card(N)} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1]$$

	A	B	C	D
A	42	127	19	3
B	10	407	138	6
C	1	122	413	50
D	4	13	56	589

TABLE 4 – Confusion matrix of logistic regression using Lasso

The error of prediction equals to $0,2745\%$. Optimal λ equals to 0.0002056876 obtained by cross validation.

II.3.5 Conclusion

Globally it's like the linear model. We don't observe great differences between the two methods.

II.4 Support-vector machine (SVM)

SVM constructs hyperplans to separate some groups of data. Thanks to the kernel trick SVM can perform non linear classification. In this part, we will perform a regression and a classification with different kernels(linear,polynomial, radial,sigmoid). The kernel trick works thanks to the Reproducing Hilbert Space (RKHS). The parameter to optimize here is $c \in \mathbb{R}^n$, where c is defined in the optimization formula :

Minimize with respect to (w, b, ξ) $\quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$ such that

$$y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i \forall i$$
$$\xi_i \geq 0$$

In the case where we use the kernel trick, we replace $\langle w, x_i\rangle$ by $\langle \phi(w), \phi(x_i)\rangle = k(w, x_i)$.
C > 0 is a tuning parameter of the SVM algorithm. It will determine the tolerance to misclassifications. If C increases, the number of misclassified points decreases, and if C decreases, the number of misclassified points increases. C is generally calibrated by cross-validation.

II.4.1 SVM with linear kernel

The Linear kernel is the most simple kernel. It predicts linearly the model. The kernel is the most common scalar product.

	A	B	C	D
A	63	7	0	5
B	127	387	103	12
C	21	169	398	34
D	2	9	67	596

(a) Confusion matrix linear kernel SVM regression

	A	B	C	D
A	0	0	0	0
B	169	402	105	17
C	19	152	428	53
D	3	7	53	592

(b) Confusion matrix linear kernel SVM classification

TABLE 5 – Confusion matrices of Linear Kernel SVM

For table 5a referring to the regression linear kernel, the prediction error equals to 27.8% while the table 5b referring to the classification with a linear kernel, the prediction error equals to 28.4%. The method using the classification cannot predict any musics noted A.

II.4.2 Polynomial SVM degree 2

A polynomial svm uses as kernel :

$$k\left(x, x'\right) = \left(1 + \left\langle x, x'\right\rangle\right)^2$$

where $p > 1$ and c a coefficient to optimize. We computed a polynomial SVM for degree 2 and degree 3. As, we observed previously the model tends to be linear. Thus if the degree is too high the predictor error will be high too.

	A	B	C	D
A	1	0	0	0
B	156	395	99	13
C	26	159	407	54
D	2	14	71	594

(a) Confusion matrix without optimization

	A	B	C	D
A	3	0	0	0
B	165	392	93	16
C	21	162	442	62
D	2	7	51	584

(b) Confusion matrix with optimization

TABLE 6 – Confusion matrices of Polynomial SVM

We obtain the prediction error without optimization and with optimization are respectively 32.4% and 28.95%. As the linear svm, the model has some difficulties to predict excellent musics.

	A	B	C	D
A	27	5	0	3
B	162	369	106	15
C	20	188	400	76
D	4	10	62	553

(a) Confusion matrix without optimization

	A	B	C	D
A	89	25	0	9
B	99	358	87	8
C	22	178	401	45
D	3	11	80	585

(b) Confusion matrix with optimization

TABLE 7 – Confusion matrices of Polynomial SVM

We obtain the prediction error without optimization and with optimization are respectively 32.55% and 28.35%.

II.4.3 Polynomial SVM degree 3

A polynomial svm degree 3 uses as kernel :

$$k\left(x, x'\right) = \left(1 + \left\langle x, x'\right\rangle\right)^3$$

	A	B	C	D
A	2	0	0	0
B	160	320	57	18
C	26	235	492	121
D	3	6	27	523

(a) Confusion matrix without optimization

	A	B	C	D
A	57	16	0	0
B	111	362	86	9
C	21	175	447	66
D	2	8	53	579

(b) Confusion matrix with optimization

TABLE 8 – Confusion matrices of Polynomial SVM

We obtain the prediction error without optimization and with optimization are respectively 33.15% and 27.75%.

	A	B	C	D
A	53	25	0	7
B	131	322	54	10
C	27	222	478	119
D	2	3	36	511

(a) Confusion matrix without optimization

	A	B	C	D
A	87	20	0	9
B	102	376	105	7
C	21	164	387	36
D	3	12	76	595

(b) Confusion matrix with optimization

TABLE 9 – Confusion matrices of Polynomial SVM

We obtain the prediction error without optimization and with optimization are respectively 31.8% and 27.75%.

The polynomial kernel of degree 3 is slightly better than the polynomial kernel of degree 2 to predict the notes of the musics.

II.4.4 Gaussian Kernel SVM

With Gaussian kernel SVM, we will perform with relaxed constraints with and without. optimization. The kernel is :

$$k\left(x, x'\right) = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

	A	B	C	D
A	10	1	0	0
B	159	390	95	16
C	19	163	435	57
D	3	7	56	589

(a) Confusion matrix without optimization (classification)

	A	B	C	D
A	41	12	0	4
B	128	370	94	11
C	20	171	430	52
D	2	8	62	595

(b) Confusion matrix with optimization (classification)

TABLE 10 – Confusion matrices of Gaussian Kernel SVM (classification)

We obtain the prediction error without optimization and with optimization are respectively 28.8% and 28.2%.

	A	B	C	D
A	80	18	0	8
B	110	378	113	8
C	21	166	381	36
D	2	10	74	595

(a) Confusion matrix without optimization (regression)

	A	B	C	D
A	72	22	0	6
B	118	382	123	8
C	21	157	380	53
D	2	11	65	580

(b) Confusion matrix with optimization (regression)

TABLE 11 – confusion matrices of Gaussian Kernel SVM (regression)

We obtain the prediction error without optimization and with optimization are respectively 28.3 % and 29.3 %.

II.4.5 Sigmoid Kernel SVM

With Sigmoid kernel SVM, we will perform without/with optimization. The kernel is :

$$k\left(x, x'\right) = \tanh\left(\kappa \left\langle x, x' \right\rangle + \theta\right)$$

We computed the errors of prediction. We figure out that the errors of prediction are higher than 50%. You can check this, in the file compiled in R.

II.4.6 SVM Conclusion

The best kernel is the Gaussian kernel. Compared to the previous models,the SVM methods are less effective to predict the right popularity class.

II.5 Classification and Regression Tree

Here we are going to implement the CART method. This method consists of separating our data into two classes by choosing a variable and a threshold that will minimize the heterogeneity of the two classes obtained. By repeating this separation process we obtain what we call a tree. The heterogeneity could be computed by different methods. In the case of regression, a simple variance calculation can be used as a heterogeneity criterion, whereas in the case of classification, a calculation of the frequency of presence of the different modes is widely used

	A	B	C	D
A	68	14	0	7
B	90	356	57	5
C	30	192	457	45
D	2	10	63	604

Table 12 – Confusion matrix of Decision Tree

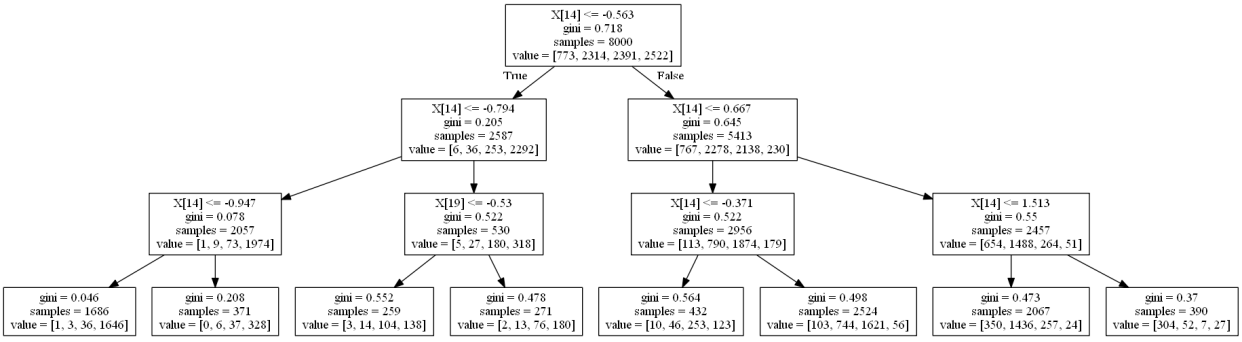


Figure 15 – Decision Tree

We obtain an accuracy score which is 0.7425.

II.6 Random Forest

Here we are going to implement the random forest method. The idea of this method is to improve the CART method seen before by fixing the variance problem.

The principle of this method is to separate our bootstrap training set and perform the CART method for each bootstrap sample. At each tree construction and for each tree node the algorithm will randomly draw m variables among the p possible ones and minimize the heterogeneity for the following nodes by considering only these m variables. It is this introduction of randomness that will allow a reduction in the variance of the model and therefore allow a better generalisation of the model. Once each tree is built for each bootstrap sample, the final tree is built by averaging the different trees obtained in the case of a regression problem or by majority vote in the case of a classification problem.

	A	B	C	D
A	45	12	0	1
B	118	384	104	12
C	26	166	409	51
D	1	10	64	597

TABLE 13 – Confusion matrix of Random Forest

We obtain an accuracy score which is 0.738. The variable importance is shown in Figure 16.

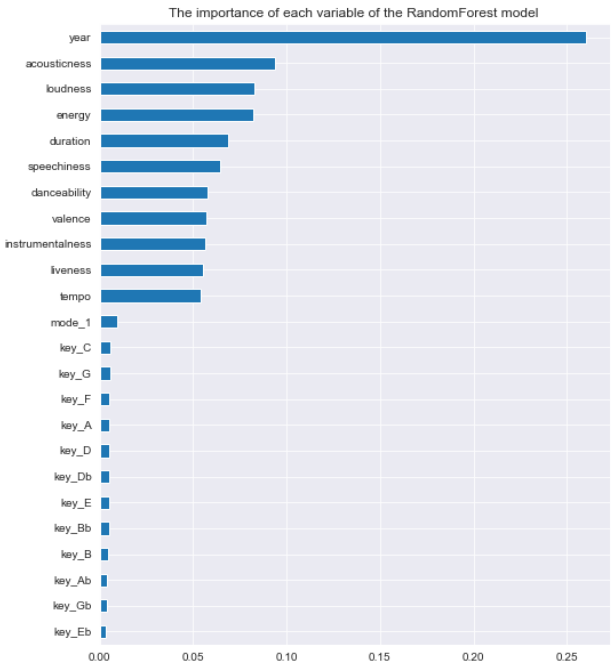


FIGURE 16 – The importance of each variable of the Random Forest model

The random forest method presents a disadvantage compared to the cart method which is the difficulty of interpretation. In order to be able to better interpret the results and in particular to conclude which variables contribute most to the construction of the model, criteria of importance of the variables have been introduced. In Figure 16 we can see that the variable year, acousticness, loudness and energy are the four main variable that contribute to the construction of our random forest. This result confirms the one found and intuited in the section on the application of elementary statistical methods.

II.7 Neural Network

In this part, we will use multi-layer perceptron with only one hidden layer. We have two ways to approach the problem :

— Classifying directly the input song by using the $softmax$ as activation function for output layer.

— Regression : the output is a real number. From there, we use the given threshold to classify songs into 4 classes.

First approach We use a MLPclassifier with only one hidden layer. Activation function for this layer is $ReLu$. We use cross-validation to find the number of neurons of the best model. The number is chosen between 5, 6, 7, 8 .The optimal number of neurons is 7. The output layer includes 4 neurons (corresponding to probability that the song belongs to 1 class $\in$ A, B, C, D), with activation function is $softmax$.

	A	B	C	D
A	72	35	0	8
B	87	357	76	4
C	30	168	445	56
D	1	12	56	593

Table 14 – Confusion matrix of Neural Network using MLPClassifier

We obtain an accuracy score 0.7335.

Second approach We use a MLPregressor with only one hidden layer. Activation function for this layer is $ReLu$. We use cross-validation to find the number of neurons of the best model. The number is chosen between 5, 6, 7, 8 .The optimal number of neurons is 7. The output is a real number and we use the given threshold to classify songs into 4 classes.

	A	B	C	D
A	49	8	0	2
B	118	439	165	10
C	22	115	359	71
D	1	10	53	578

Table 15 – Confusion matrix of Neural Network using MLPRegressor

We obtain an accuracy score 0.7. The perfomance of MLPClassifier is slightly better than MLPRegressor.

II.8 Choosing the best model

Table 16 – Table of results

	Train	Test	F1
Linear Regression without penalisation	0.7144	0.7155	0.6496
Linear Regression with Lasso penalisation	0.7138	0.717	0.6476
Linear Regression with Ridge penalisation		0.695	
Logistic Regression without penalisation	0.6837	0.685	0.5451
Logistic Regression with Lasso penalisation	0.6843	0.686	0.5451
Logistic Regression with Ridge penalisation		0.686	
Logistic Regression with eslastic penalisation		0.726	
Support Vector Machine with linear kernel	0.6672	0.671	0.5194
Support Vector Machine with polynomial degree 2 kernel		0.7165	
Support Vector Machine with polynomial degree 3 kernel		0.7225	
Support Vector Machine with Gaussian kernel	0.7423	0.703	0.5883
Decision Tree	0.7382	0.7425	0.6876
Random Forest	0.999	0.738	0.6847
Neural Network (MLPClassifier)	0.7372	0.7335	0.6724
Neural Network (MLPRegressor)	0.7188	0.7	0.6376

We summarize all the obtained results in Table 16. To evaluate all models, we add the F1 score which is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

From Table 16, we observe that
— F1 scores is distributed between 0.51 and 0.68.
— The accuracies on test set is distributed between 0.65 and 0.75
— There were no considerable differences in terms of performance among all our tested models
The accuracy score is not relatively high. This could be explained by the fact that all of our classification algorithms assume the data is linearly separable. But in reality, the data was mostly likely not linearly separable.
Using the kernel on Support Vector Machine method has a noticeable effect. The accuracy score of model increases quite a lot. In our case, polynomial kernels works better than Gaussian kernel. Decision Tree model is the best model, with the acurracy score and F1 score on test set are the highest among all our models.

Conclusion

In this project, we try to predict the popularity of songs based on their different characteristics. Two approaches are possible : classifying directly the popularity or regression, i.e predict the popularity score of songs on the scale 0-100 and then classifying them with the given thresholds. We perform these two approaches with different methods : linear regression, logistic regression, SVM, classification and Regression Tree, random Forest and neural network. For each method, we try to find the optimal corresponding parameters by cross-validation. Then for each optimized method we choose the more suitable method according to our data. It is important to highlight that there is not a general better model : one method could give better results for one type of problem and could be the worst method for another problem. In our case, we have found that globally the best methods are those resolving the classification problem. More precisely, according to the F1 criterion, the CART method seems to be most suitable.

Concerning the importance of the different variables in our problem, we have seen throughout our project that the year variable has always been the variable contributing most to the discrimination of our models and therefore to the latter's construction. Other variables seem to be determinant for this problem such as the acousticness, the loudness, or the energy while the variable concerning the primary key of the track seems not to be important in our case.

Research has been done about predicting the popularity of a song. It is then interesting to compare our results with others. For instance, students from Stanford University have already proposed a solution for this problem and have published an article online that you can find following this link : `http://cs229.stanford.edu/proj2015/140_report.pdf`. From this article we can read "Through several different feature selection algorithms, we were able to identify the most influential features in our dataset by taking the intersection among the feature selection algorithms, namely artist familiarity, loudness, year, and a number of genre tags.[...].We found that the acoustic features are not nearly as predicative.". We have found the same result as them concerning the importance of the year and the loudness. However, it seems that for them the most important variable is the namely artist familiarity : this such variable was not in our dataset hence we do not find this result. What is more surprising is that in their case they found that acousticness was not a determinant variable while for us it is one of the most important.

Finally, even if we were able to give a solution for this problem, it is one solution among several and there is no real solution. What we could say as a general result that we can find in all articles dedicated to predicting the popularity of a song is that the year and the loudness are always one of the most important variables. Moreover, as our data are not really linear or gaussian or following a certain probability rule the best machine learning method to predict will not be a basic method that assumes a certain behavior of our data : more sophisticated methods such as CART or random forest are more suitable for this problem.

INSA Toulouse
135, Avenue de Rangueil
31077 Toulouse Cedex 4 - France
www.insa-toulouse.fr

Université Fédérale Toulouse Midi-Pyrénées

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE