



Year Prediction -
Million Songs Dataset

Sources

This data is a subset of the Million Song Dataset:

<http://labrosa.ee.columbia.edu/millionsong/>

a collaboration between LabROSA (Columbia University) and The Echo Nest.



Dataset information



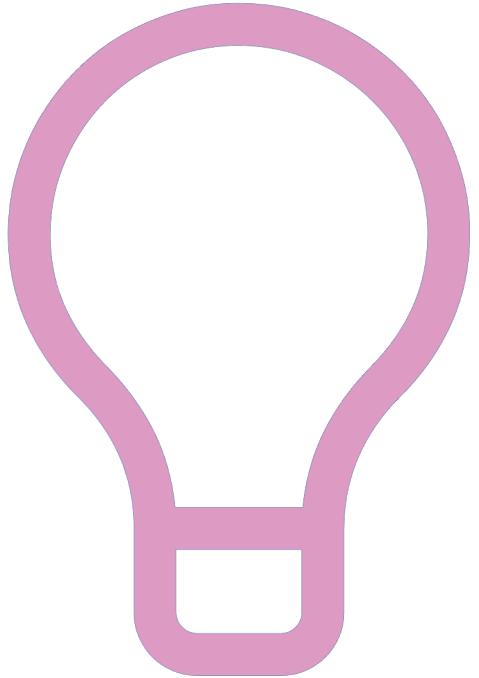
The dataset is composed of 515 345 records of songs, composed between 1920 and 2011.



Each song is characterized by 91 variables : the first feature is the year of the released of the song, the target of this dataset. The other 90 features are various timbre features related to the song audio.



In the musical world, timbre is what makes a particular musical instrument or human voice have a different sound from another, even when they play or sing the same note.

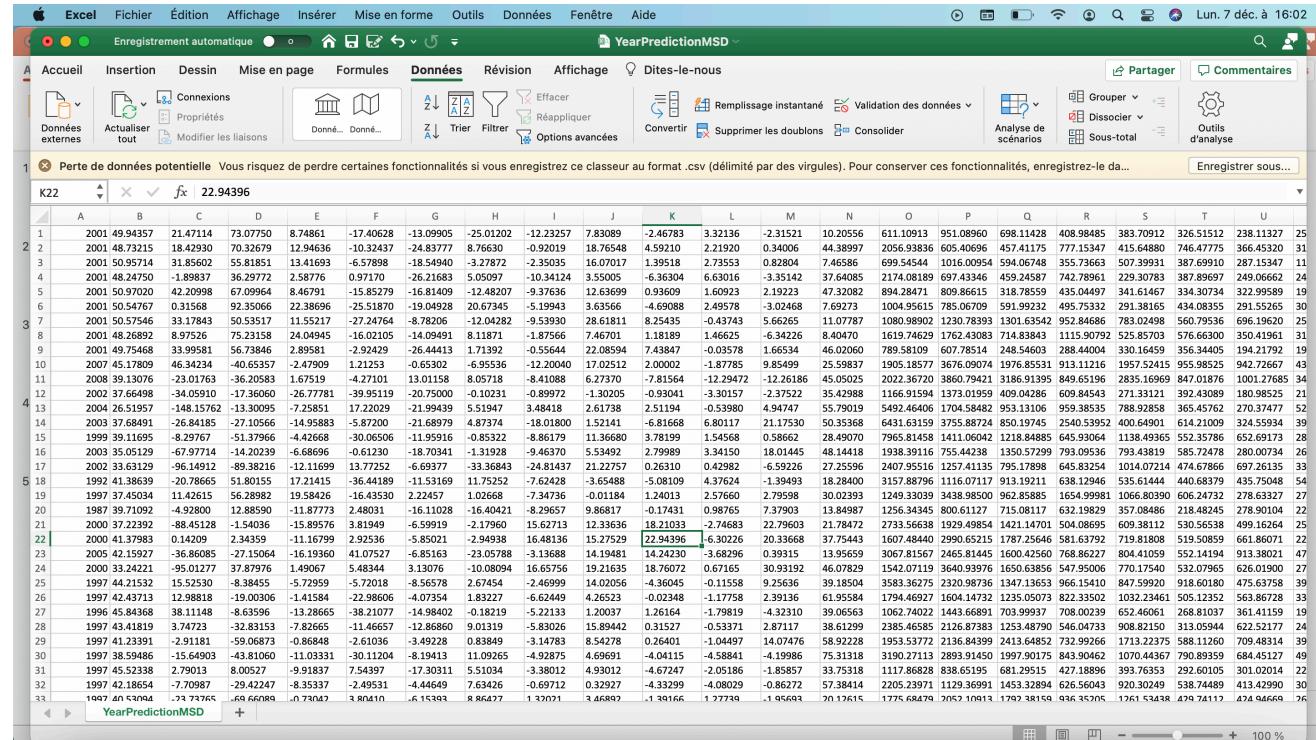


Problematic

- Is there a strong relation between the musical features of a song to the year it was composed?
- can we design several models that can predict the year from the other 90 musical features?
- If the answer is yes, which one will fit in the better way?

A positive answer to the second question would reveal a profound insight on the nature of a musical composition

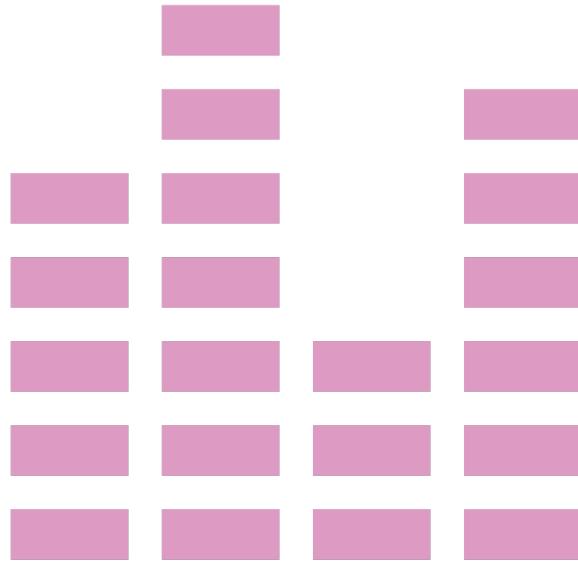
Data sourcing



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	2001	49.94357	21.47114	73.07750	8.74861	-17.40628	-13.09905	-25.01202	-12.23257	7.83089	-2.46783	3.32136	-2.31521	10.20556	611.0913	951.08960	698.11428	408.98485	383.70912	326.51512	238.11327
2	2001	48.73215	18.42930	70.32679	12.94636	-10.32437	-24.83778	8.76630	-0.92019	18.76548	4.59210	2.21920	0.34006	44.38997	2056.93836	605.40696	457.41175	77.15347	145.64880	746.47775	366.45320
3	2001	50.95714	31.85602	55.81851	13.41693	-6.57898	-18.54940	-3.27872	-2.35035	16.07017	1.39518	2.73553	0.82804	74.65886	699.54544	1016.00954	594.06748	355.73663	507.39391	387.69910	287.15347
4	2001	48.24750	-1.89837	36.29772	2.58776	0.97170	-26.21683	5.05097	-10.34124	3.55005	-6.36304	6.63016	-3.35142	37.64085	2174.08189	697.43346	459.24587	742.78961	229.30783	387.89697	249.06662
5	2001	50.70720	42.20998	67.09964	8.46791	-15.85279	-16.81409	-12.48207	-9.37636	12.63699	0.93609	1.60923	2.19223	47.32082	894.28471	809.86615	318.78559	435.04497	341.61467	334.30734	322.99589
6	2001	50.54767	0.31568	92.30566	22.38696	-25.51870	-19.04928	20.67345	-5.19943	3.63566	-4.69084	2.49578	-3.02468	7.69273	1004.95615	785.06709	591.99232	495.75332	291.38165	434.08355	291.55265
7	2001	50.7546	33.17843	50.53517	11.55217	-27.24764	-8.78206	-12.04282	-9.53930	28.61811	8.25453	-0.43743	5.66265	11.07787	1080.98902	1230.78393	1301.63545	952.84686	783.02494	560.7953	696.19620
8	2001	48.26892	8.97526	75.23158	24.04945	-16.02105	14.09491	8.11871	-1.87566	7.46701	1.18189	1.46625	-6.34226	8.40470	1619.74629	1762.43083	714.83843	1115.07972	525.85703	576.63630	350.41961
9	2001	49.75468	33.99581	56.73846	2.89581	-2.92429	-26.44413	1.71392	-0.55644	7.43847	-0.03578	1.66534	46.02060	789.58109	607.78514	248.54603	388.44004	330.16459	356.34405	194.21792	
10	2007	45.75802	46.43424	-40.65357	-2.47909	1.21253	-0.65302	6.59536	-12.20040	17.02512	2.00002	-1.87785	9.85949	25.59837	1905.18577	3676.09074	1976.85531	1957.52415	952.74267	43	
11	2009	39.13039	-23.01763	-36.20582	1.67519	-4.77101	12.01158	9.05718	8.41088	6.27370	-8.71564	-12.39472	-12.26186	45.05005	2022.36720	3860.79421	3186.91399	849.65196	3825.16699	847.01876	1001.37768
12	2002	37.66498	-34.0510	-17.36060	-26.77781	-39.55119	-20.75000	0.10231	-0.89972	-1.30205	-0.93041	-3.30157	-2.37522	3.42988	1166.91594	1373.01959	409.42836	609.84543	271.33121	392.43089	180.98525
13	2004	26.51957	149.15762	13.30095	-7.25851	17.22029	21.99493	5.51947	3.48418	2.61738	2.51194	-0.53980	4.94747	55.79019	5492.46406	1704.58482	953.13106	958.38535	788.92858	365.45762	270.37477
14	2003	37.68491	-26.84185	-27.10566	14.95883	-5.87200	-21.68979	4.87374	-18.01800	1.52141	-6.81668	6.80117	21.17530	50.35368	6431.63159	3755.88724	850.19745	2540.53952	400.64901	614.21009	324.55934
15	1999	39.11695	-8.29767	-51.37966	-4.42668	-30.06506	-8.86179	11.36680	-8.86179	3.15468	0.58662	28.49070	7965.81458	1411.06042	1218.84885	645.93064	1138.49365	552.85786	652.69173	28	
16	2003	35.05129	-67.97714	-14.20239	-6.68696	-0.61230	-18.70341	-1.31928	-9.46370	5.53492	2.79989	18.01445	48.14418	198.39116	755.57299	793.09536	793.43819	585.24748	280.00734	26	
17	2002	33.63129	-96.14912	-89.38216	-12.11699	13.77252	-6.69377	-33.36843	-24.81437	21.22757	0.26310	4.02982	-6.59226	27.25596	2407.95516	1257.41135	795.17988	645.83254	1014.07214	474.67868	697.26135
18	1992	41.38634	-20.78665	51.80155	17.21415	-36.44189	-11.51369	11.75252	-7.62428	-3.65488	-0.58109	4.37624	-1.39493	18.28400	3157.88796	1116.07117	913.19211	638.12946	535.61442	440.83879	435.75048
19	1997	37.40534	11.42615	56.28982	19.58426	-16.43530	2.22457	1.02661	-7.34736	-0.01184	1.24013	2.57660	2.79598	30.02393	1249.33039	3438.98500	962.85885	1654.99981	1066.80390	606.24732	278.63327
20	1987	39.71092	-4.92800	12.88590	-11.87773	2.48031	-16.11028	-16.40421	-8.29657	9.98617	-0.17431	0.98765	2.73903	13.84987	1256.34345	800.61227	715.08117	632.19829	357.08486	218.48245	290.91004
21	2000	37.22392	-88.45128	-1.54036	-15.89576	3.81949	-6.59919	-2.17960	15.62713	12.33366	18.21033	-2.74683	22.79603	21.78472	2733.56638	1929.49854	1421.14701	504.08695	609.38112	530.55638	499.16264
22	2000	41.3798	0.14209	2.34359	-11.16799	2.92536	-5.85021	-2.94938	16.48136	15.27529	22.94396	6.30226	20.33668	37.75443	1607.48440	2990.65215	1787.25641	581.63792	719.81800	519.50859	661.86071
23	2005	42.15927	-36.86085	-27.15064	-16.19366	41.07527	-6.85163	-23.05788	-3.13688	14.19481	14.24230	-6.68296	0.39315	13.95659	3067.81567	2465.81445	1602.42560	768.86227	804.41059	552.14194	913.38021
24	2000	33.24221	-95.01277	37.87976	1.49067	5.48344	3.13076	-10.08094	16.65756	19.21635	18.76072	0.67165	30.93192	46.07829	1542.07119	3640.93976	1650.63850	547.95006	730.17540	532.07996	626.01900
25	1997	44.21532	15.52530	-8.38455	-5.72959	-5.72018	-8.56578	2.67454	-2.46999	14.02056	-4.36045	-0.11558	9.25630	39.18504	3583.36275	2320.98738	1347.13653	966.15410	847.59920	918.60108	475.63758
26	1997	42.43713	12.98818	-19.00306	-1.41584	-22.98602	-4.07354	1.83227	-6.62449	4.26523	-0.02348	-1.17758	2.39136	61.95584	1794.46927	1604.14732	1235.05073	822.33502	1032.23461	505.12325	563.86728
27	1996	45.84368	38.11148	-8.63596	-13.28665	-38.21077	-14.98402	-0.18219	-5.22133	1.20037	1.26164	-1.79819	-4.32310	39.06563	1062.74022	1443.66891	703.99937	708.00239	652.46061	268.81037	361.41159
28	1997	43.41819	3.74723	-32.83153	-7.82665	-11.46657	-12.86860	9.01319	-5.83026	15.89442	0.31527	-0.53371	2.87117	38.61299	2385.46585	2126.87383	1253.48790	546.04733	908.82150	313.05944	625.52177
29	1997	41.23391	-2.91181	-59.06873	-0.86848	-2.61036	-3.49228	0.83849	-3.14783	8.54278	0.26401	-1.04497	14.04746	58.92228	1953.53772	2136.84399	2413.64852	732.99266	1713.22375	588.1260	709.48314
30	1997	38.59486	-15.64903	-43.81060	-11.03331	-30.11204	-8.19413	11.92625	-4.92875	4.69691	-4.04115	-4.58841	-4.19986	75.31318	3190.27113	2893.91450	1997.90178	843.90462	1070.44367	790.89359	684.45127
31	1997	45.52338	2.79013	8.00527	-9.91837	7.54397	5.51034	-3.38012	4.93012	-4.67247	-2.05186	-1.85857	17.75318	1117.86828	838.65195	681.25915	427.18896	393.76353	292.60105	301.02014	
32	1997	42.18654	-7.70987	-29.42427	-8.35337	-2.49531	-4.46446	7.63426	-0.69712	0.32927	-4.33299	-0.86272	57.38414	205.23971	1129.36991	1453.32894	626.56043	920.30249	538.74489	413.42990	
33	1997	40.53094	-2.727365	-69.66098	-0.73047	3.80410	-6.15393	8.86427	-1.37021	3.46892	-1.39166	1.27739	-1.95693	20.12615	1775.68479	2052.10913	1792.38159	936.35705	1261.53438	479.74112	474.04669

- Here is a quick look of the dataset we can find on UCI's website :

Data sourcing



- Therefore we have to create a list which will contain the names of the different variable. We said before that the variables are timbres from the songs
- As we are not musical experts, the 90 features are of no interest to us, so we will simply name them: timbre1, timbre2, timbre3, ..., timbre90.
- So after some manipulation, our dataset will look like that :

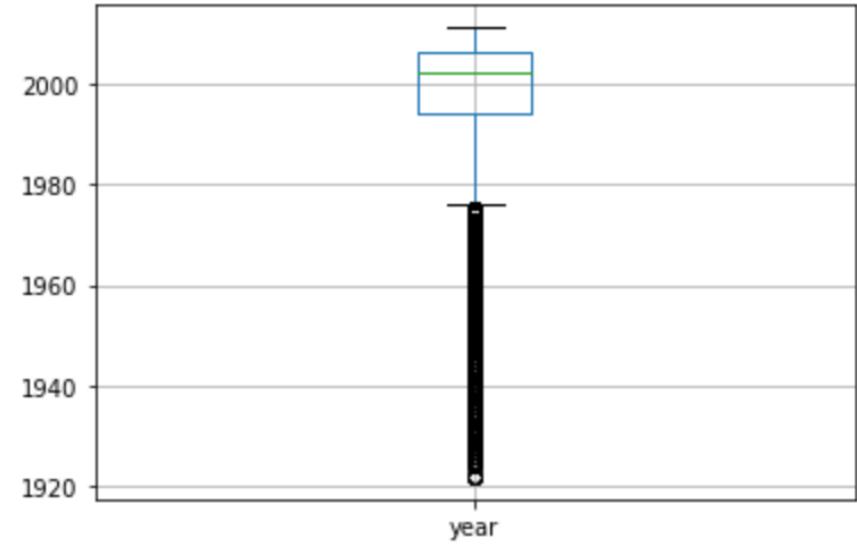
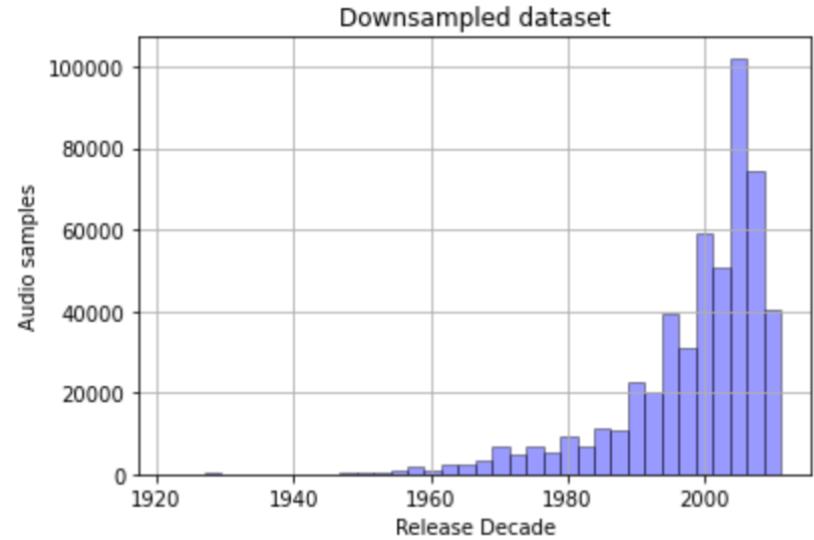
Out[47]:

	year	timbre1	timbre2	timbre3	timbre4	timbre5	timbre6	timbre7	timbre8	timbre9	...	timbre81	timbre82	timbre83	timbre84
0	2001	1.080575	0.391265	1.826532	0.464657	-0.474730	-0.278204	-1.552371	-1.310845	0.387704	...	-0.085335	0.108508	0.142775	-0.237355
1	2001	0.880919	0.332292	1.748539	0.721828	-0.164945	-1.191173	0.765681	0.109626	1.420941	...	-0.314250	0.306236	-0.069483	0.052017
2	2001	1.247622	0.592600	1.337173	0.750657	-0.001110	-0.702100	-0.060914	-0.069956	1.166254	...	-0.396186	0.566683	-0.756534	-0.284019
3	2001	0.801044	-0.061805	0.783683	0.087218	0.329180	-1.298429	0.510714	-1.073355	-0.016803	...	0.586237	-0.559427	-0.478689	-0.890161
4	2001	1.249775	0.793334	1.657037	0.447460	-0.406775	-0.567138	-0.692498	-0.952197	0.841844	...	-0.181585	0.099672	0.191319	-0.585576

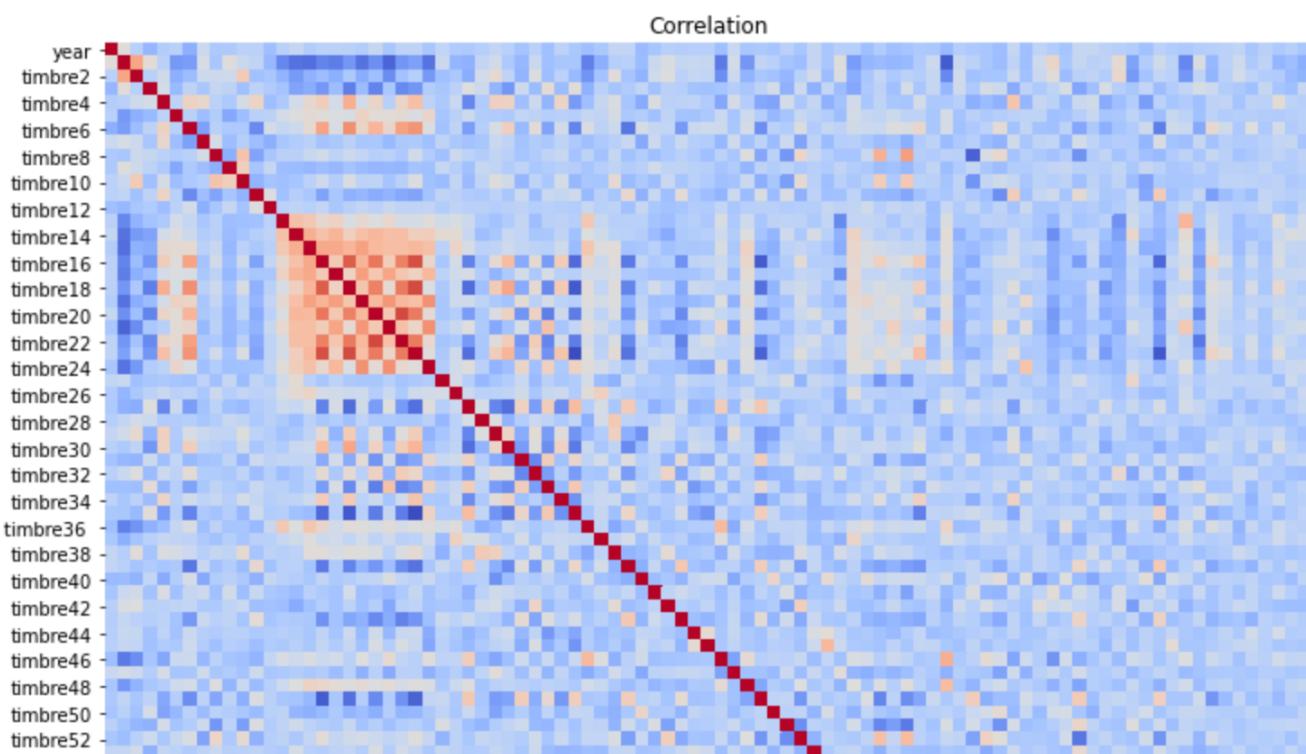
5 rows × 91 columns

EDA

- We will first take a look at our target :
- We notice that the frequency of the counts start to take off in the late 1970s to dawn of 1980s, likely coinciding with the invention of the CDs by Philips and Sony! And we also can clearly see is that the central mass between the 1st and 3rd quartiles lies in years ranging from the mid 1990s to the mid 2000s.

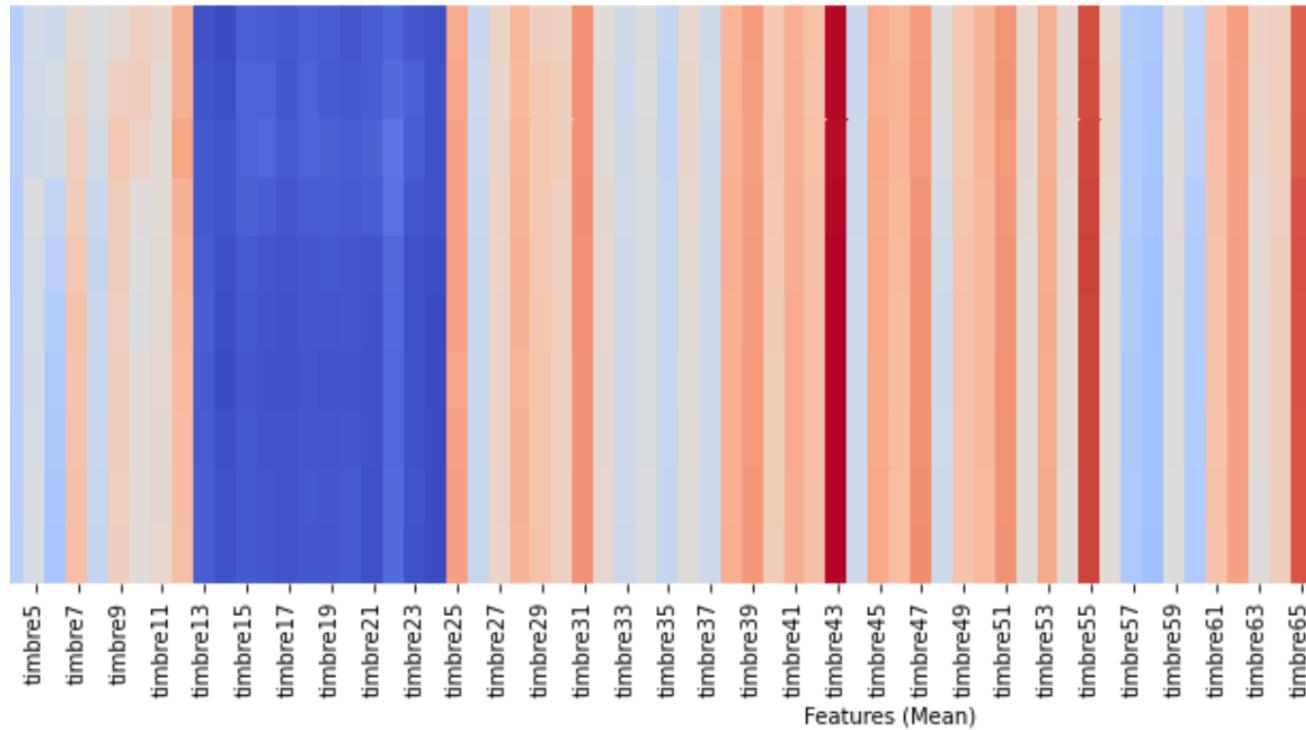


EDA



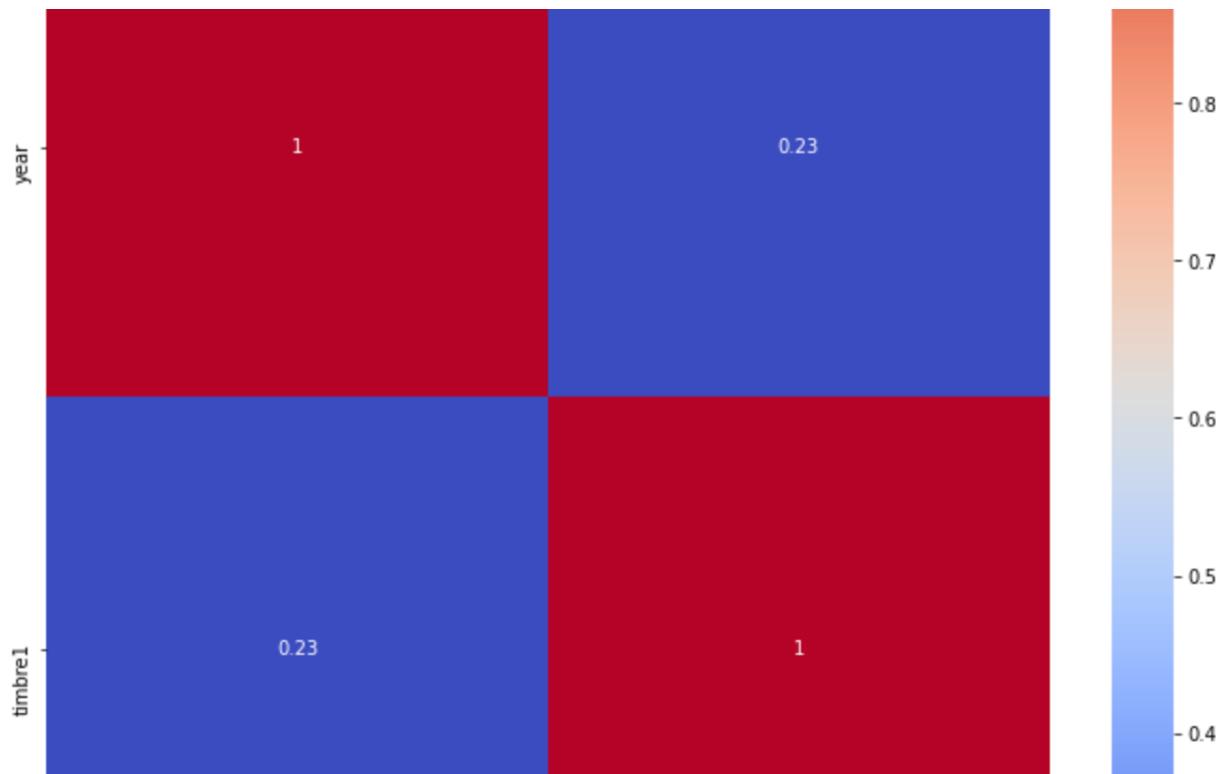
- Let's see if there is a correlation between the features and the target
- We start by scaling the data with StandardScaler method, that set the mean on 0 and the std on 1.
- Let's plot the correlation matrix :
- We notice some important correlation around the variables timbre14 to timbre24, but it is not really helpful concerning the target of our dataset

EDA



- We can see that except for the variable 'timbre1','timbre2' the features are pretty much the same by release decade...
- Then we made a graph that shows how the features differ by release decade. We applied the MinMaxScaler for this graph and we grouped the data by group of tens year for the graph to be more legible.

EDA



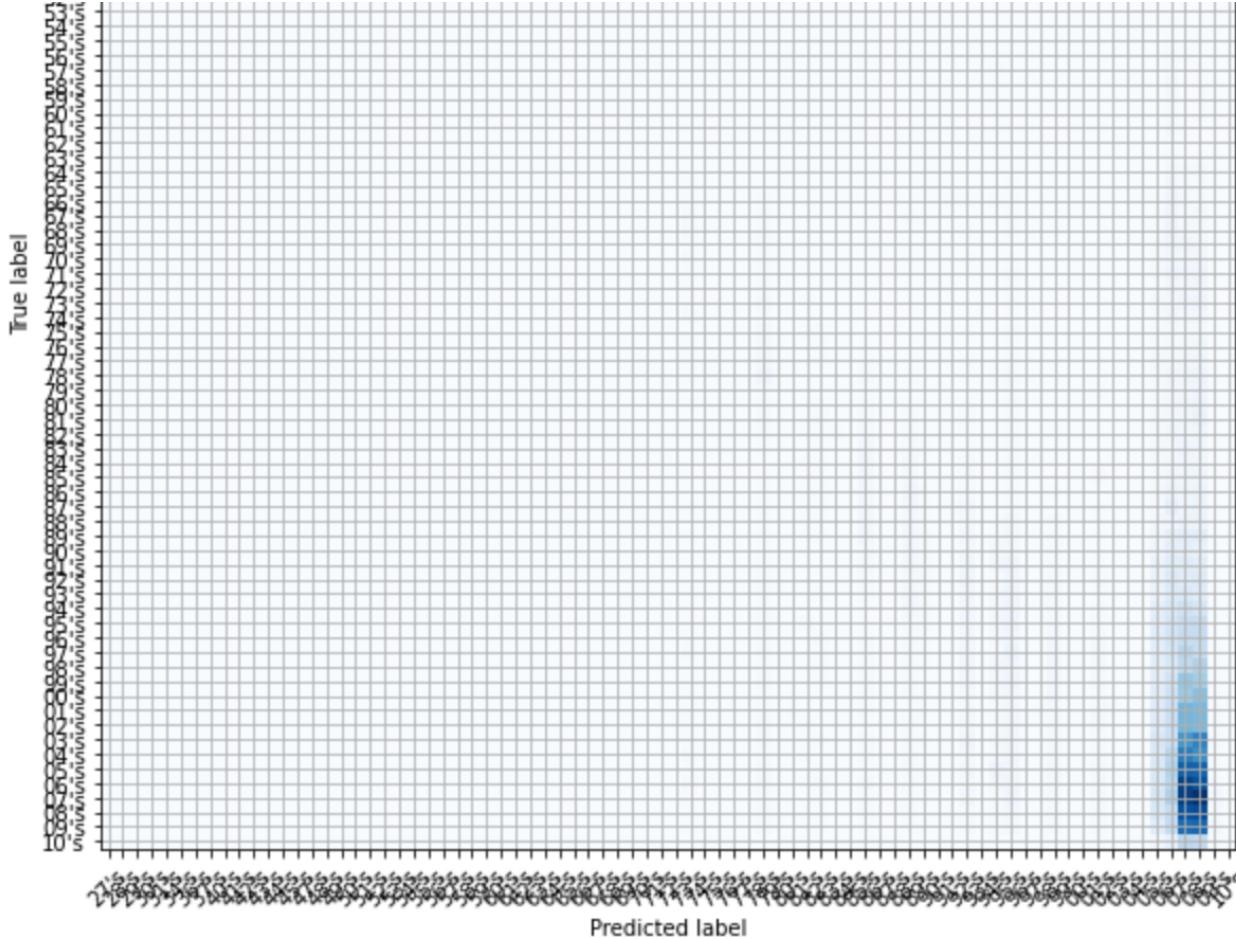
- And after taking a closed look on the correlation between the variable 'timbre1' and our target, the correlation appear to be only 0,23

Preprocessing the data

The first step is to separate our data into training and test set. We followed the guide of the UCI's website that told us to take the first 463715 data for the training set and the rest for the test set. So, we have this two format of datasets :

```
Train : X (463715, 90) , Y (463715,)  
Test : X (51630, 90) , Y (51630,)
```

Model



We tried several models :

- Logistic regression
- Decision tree
- Neural network

But the accuracies did not exceed 0,1 and the confusion matrix made with the result of the neural network was not really satisfying...

Model

So, we tried something else :

We created a second dataset where the target were grouped by decade. We scale this dataset like the other : with the standardscaler method

From this dataset we did the same split as earlier : 463175 data for the training set and the rest for the test set

Then we focused our work on the neural network

Model – Neural network

Because we wanted to have to best model possible, we made two grid search.

The first one were used to find the better lost function and optimizer.

A loss function is used to optimize the parameter values in a neural network model. Loss functions map a set of parameter values for the network onto a scalar value that indicates how well those parameter accomplish the task the network is intended to do. And the optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.

In this case, the binary cross entropy seems to be the better loss function and the Adam optimizer were chosen by the algorithm.

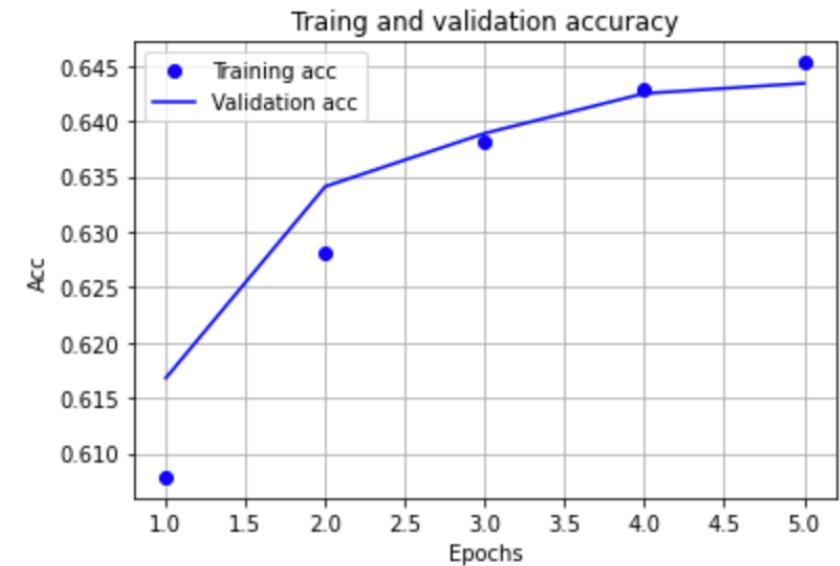
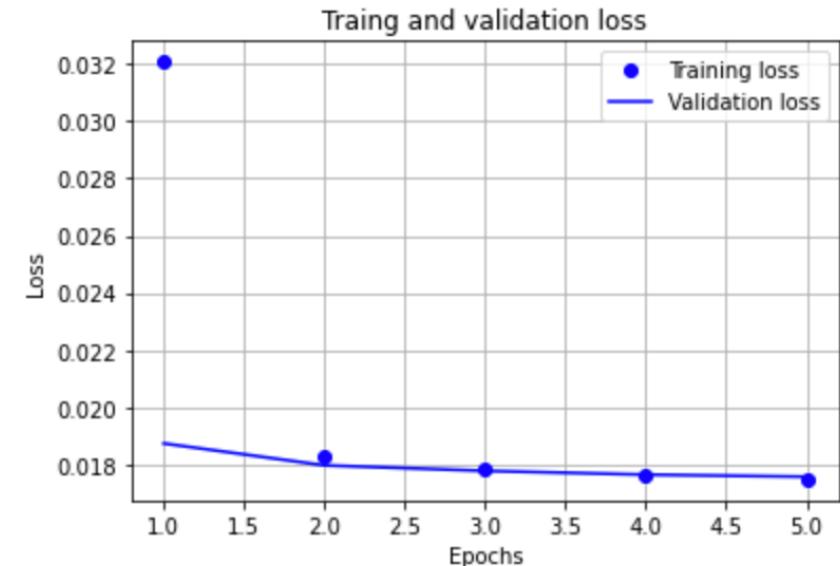
The second grid search were for the activation function (who is the function that is applied at each neuron) and the better one appear to be the tanh function.

Model - neural network

So, we trained the model with the best parameters and we plotted the loss function and the accuracy function.

The loss training function and validation function are quite the same which is good, and it clearly is decreasing

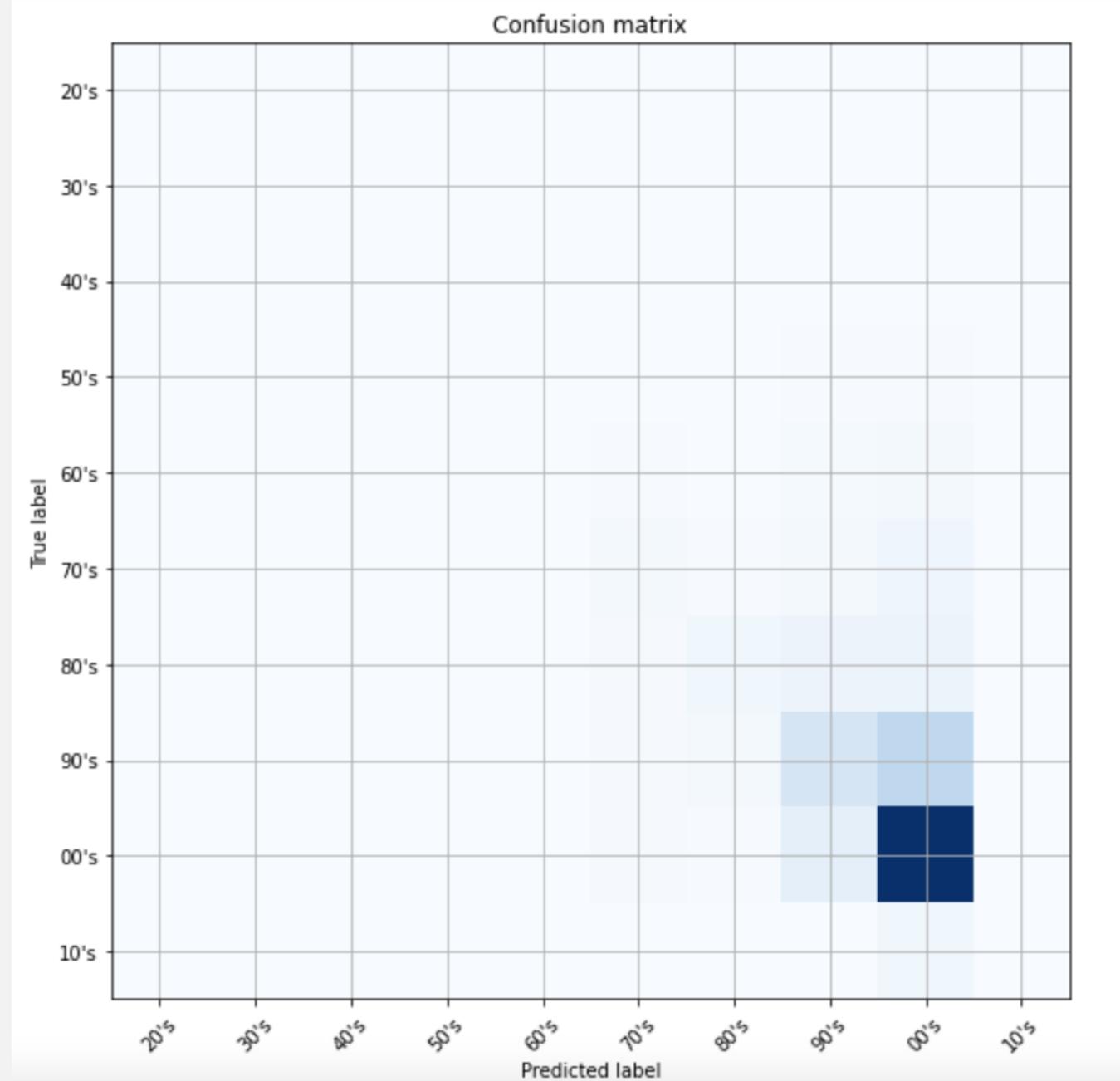
The accuracy training function and validation function are quite the same too and keep increasing.



Model - neural network

We have a final accuracy of 0.64, which is way better than earlier. And here is the confusion matrix : →

We can see from this confusion matrix that the algorithm only guess right the song from the 2000 decade. Because a big part of our dataset is composed by song from this decade, it's difficult for the algorithm to guess right the other songs.



Model - Conclusion

To finish we applied the logistic regression to our new dataset and the accuracy were pretty closed to the one from the neural network.

Here is a graph to compare the different model that we work one :

The best model is definitely the neural network with the grouped data

