Year Prediction - Million Songs Dataset

Sources

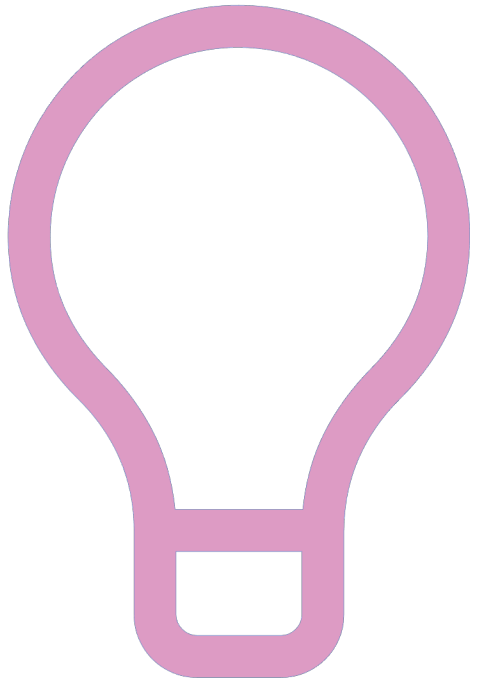
This data is a subset of the Million Song Dataset: http://labrosa.ee.columbia.edu/millionsong/
a collaboration between LabROSA (Columbia University) and The Echo Nest.

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.

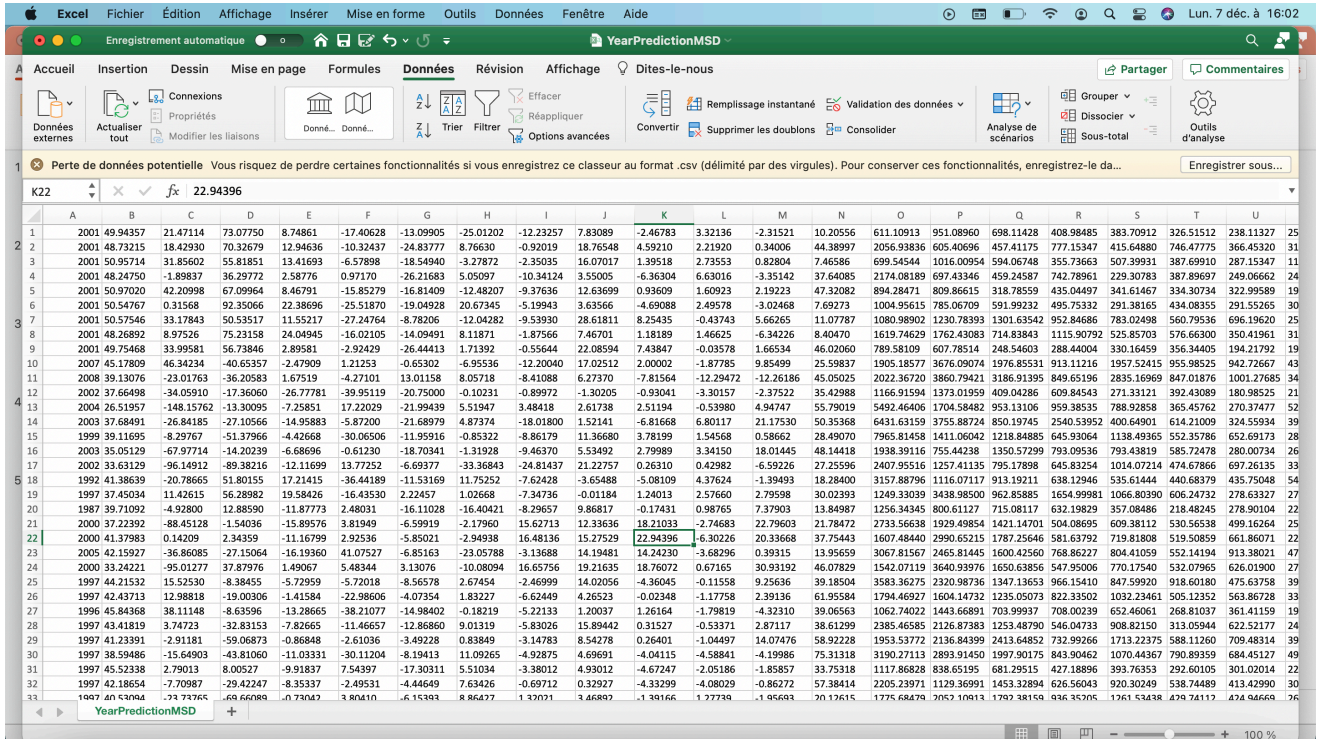


Problematic

- 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



- 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 :

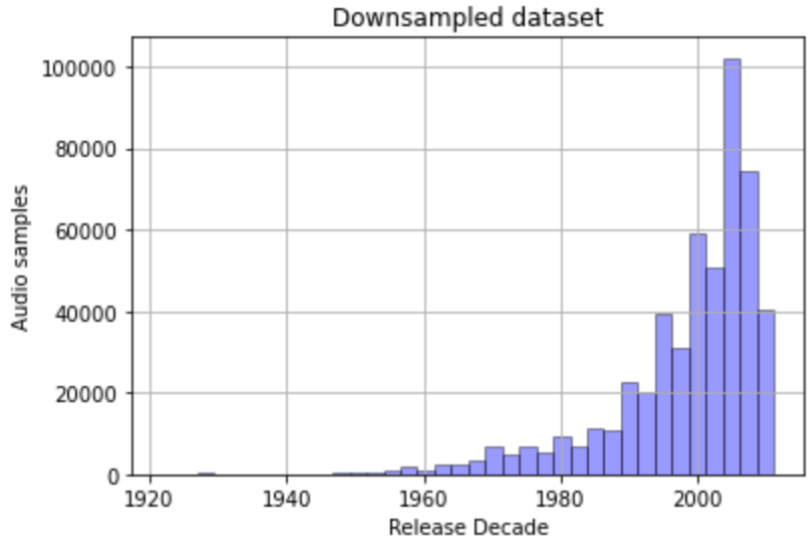
Out[47]:

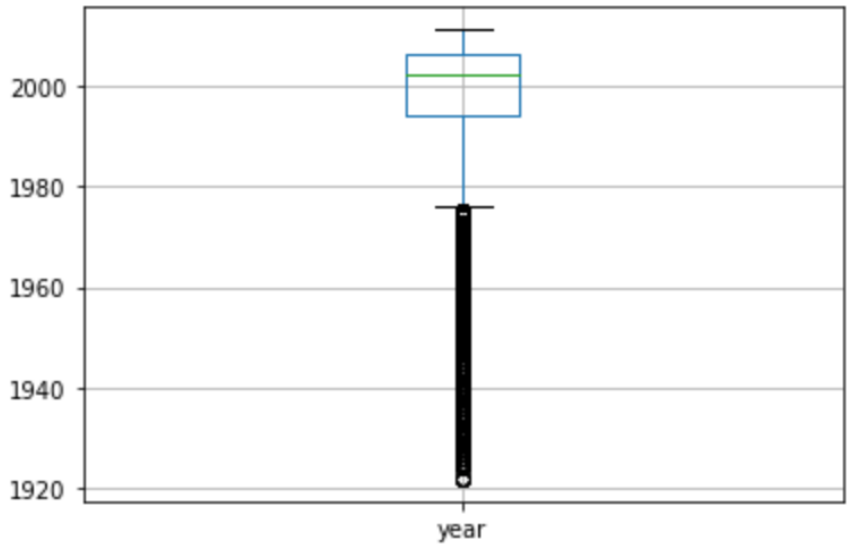
	year	timbre1	timbre2	timbre3	timbre4	timbre5	timbre6	timbre7	timbre8	timbre9	...	timbre81	timbre82	timbre83	timbre84
0	2001	1.080575	0.391265	1.826532	0.464657	-0.474730	-0.278204	-1.552371	-1.310845	0.387704	...	-0.085335	0.108508	0.142775	-0.237355
1	2001	0.880919	0.332292	1.748539	0.721828	-0.164945	-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

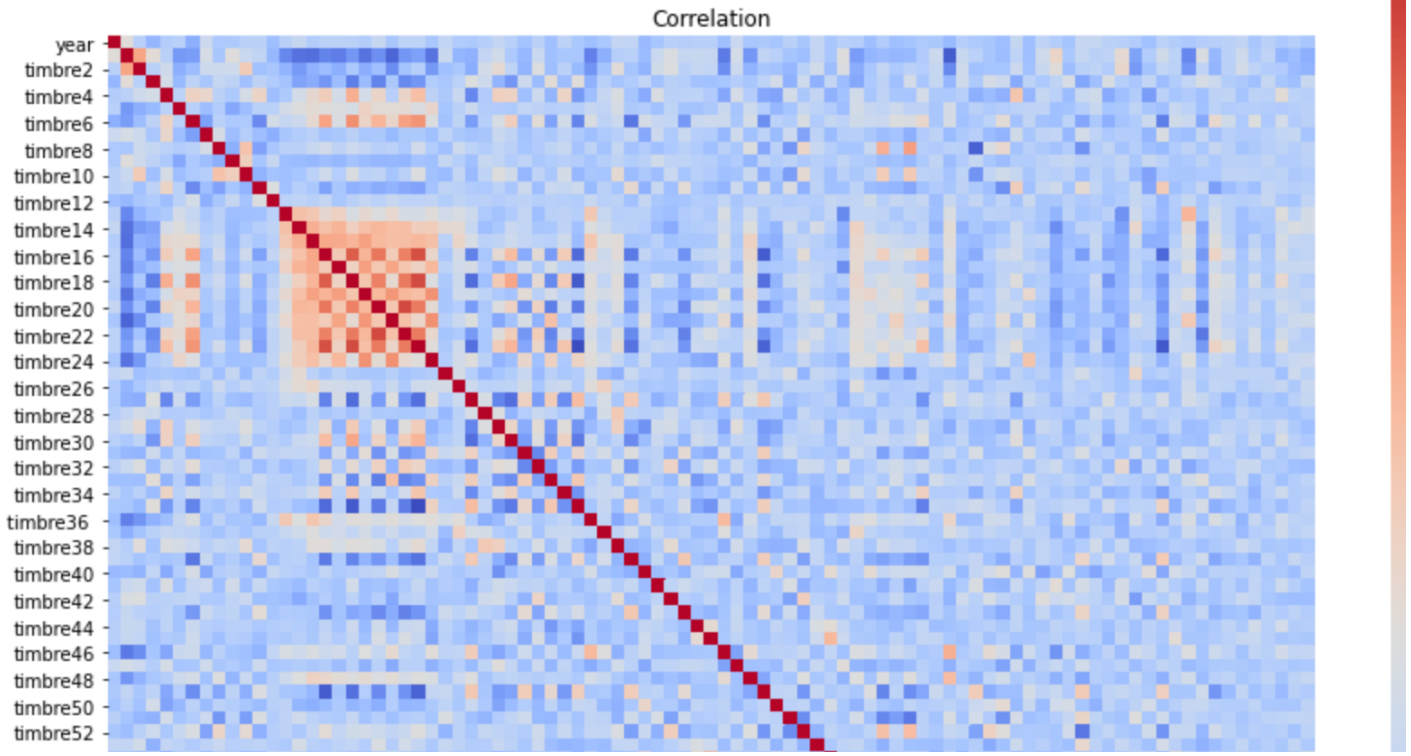
EDA

- 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.





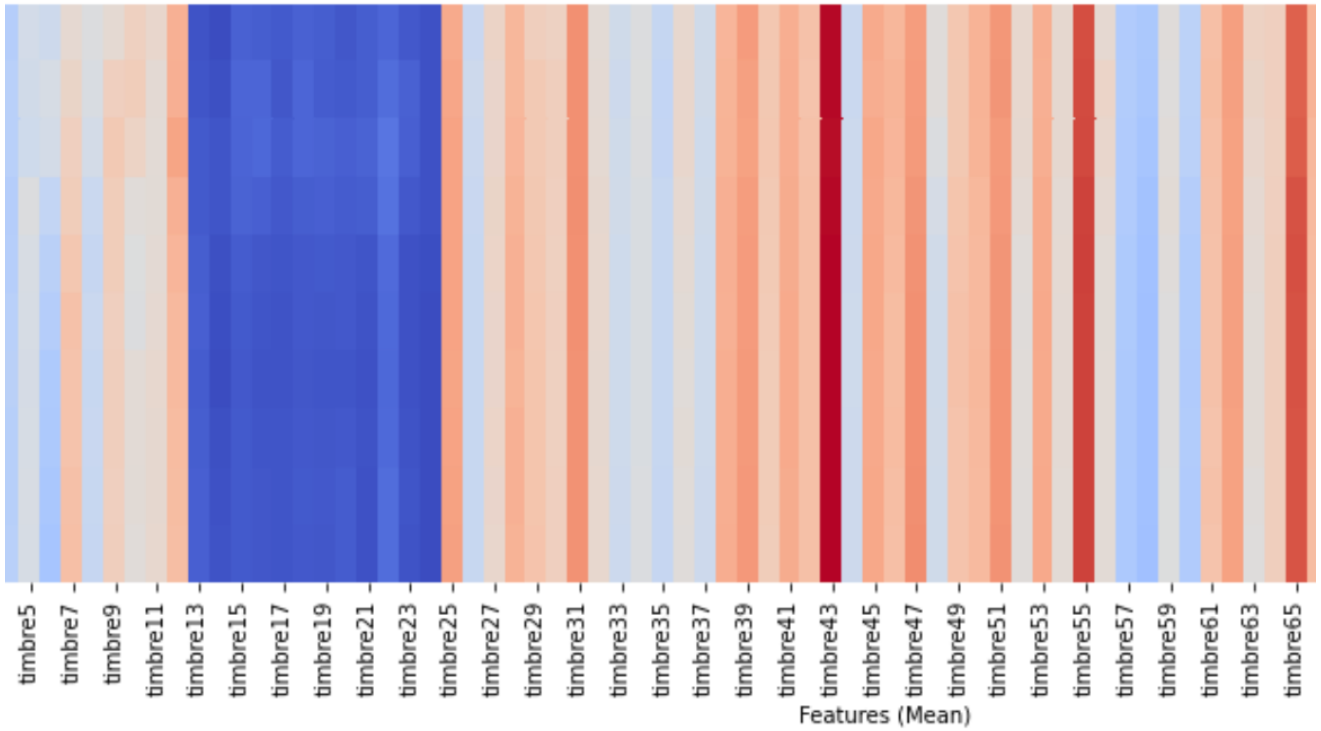
EDA



- Let's see if there is a correlation between the features and the target
- We start by sclaing 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 timbre timbre24, but it is not really helpful concerning the target of our dataset

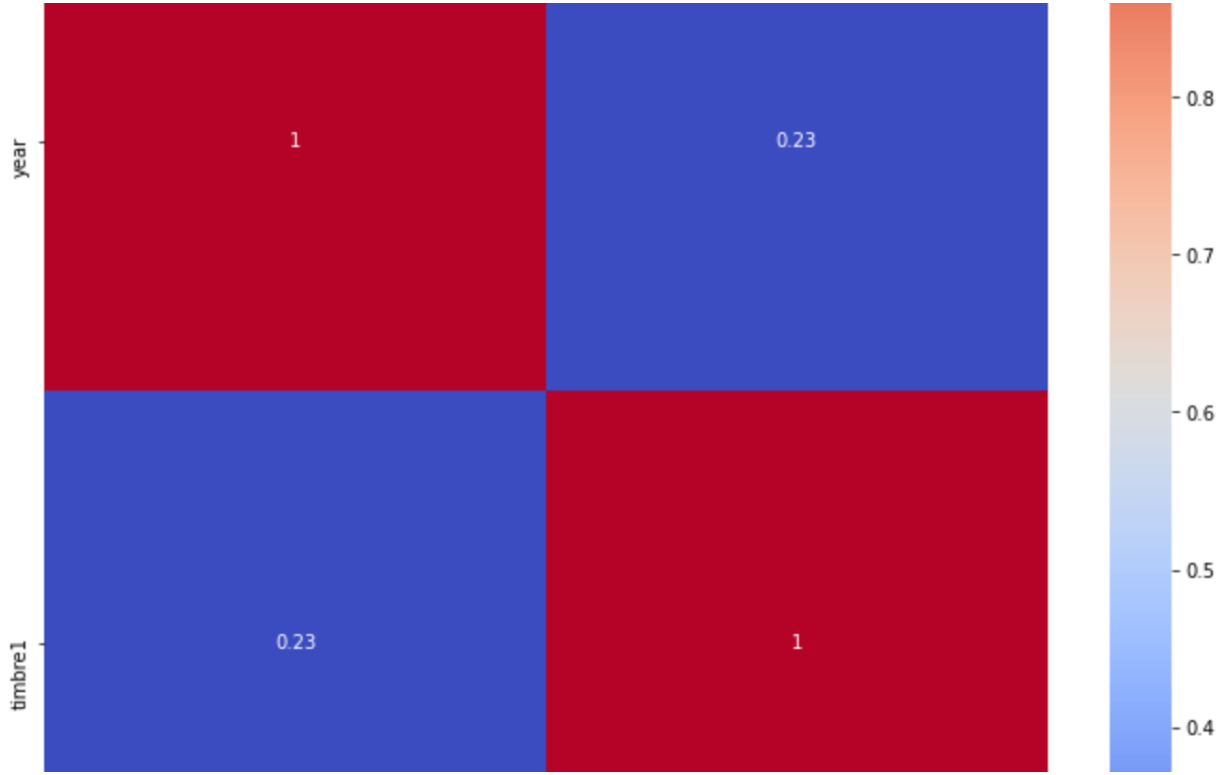
EDA



- 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.

- We can see that except for the variable 'timbre1','timbre2' the features are pretty much the same by release decade...

EDA



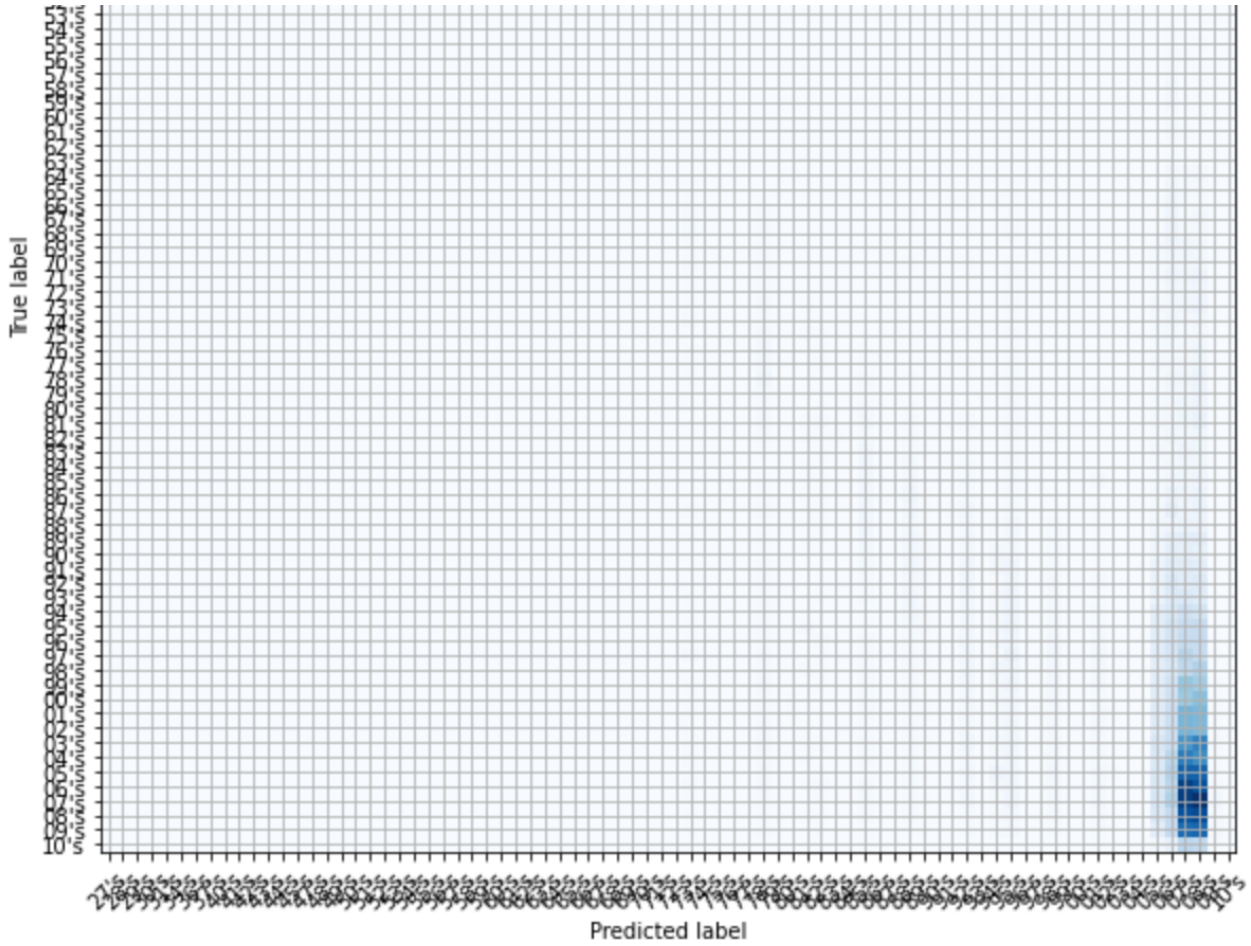
- And after taking a closed look on the correlation between the variable ‘timbre1’ and our target, the correlation appear to be only 0,23 ….

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.

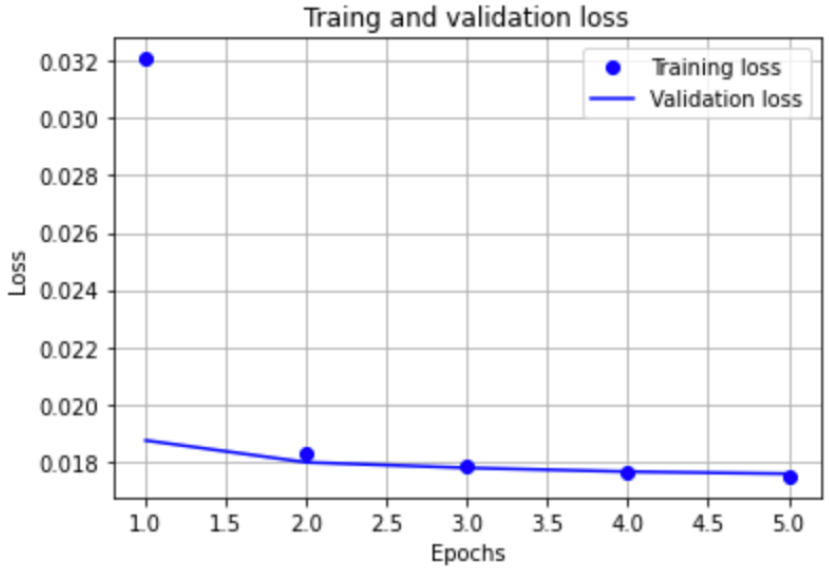
The second grid search were for the activation function (who is the function that is applied at each neuron) and the better one appear to be the tanh function.

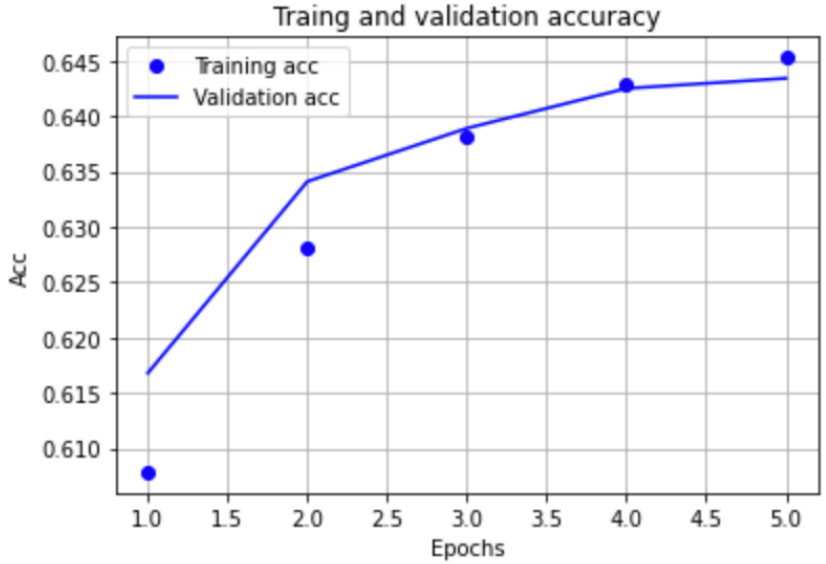
Model - neural network

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.

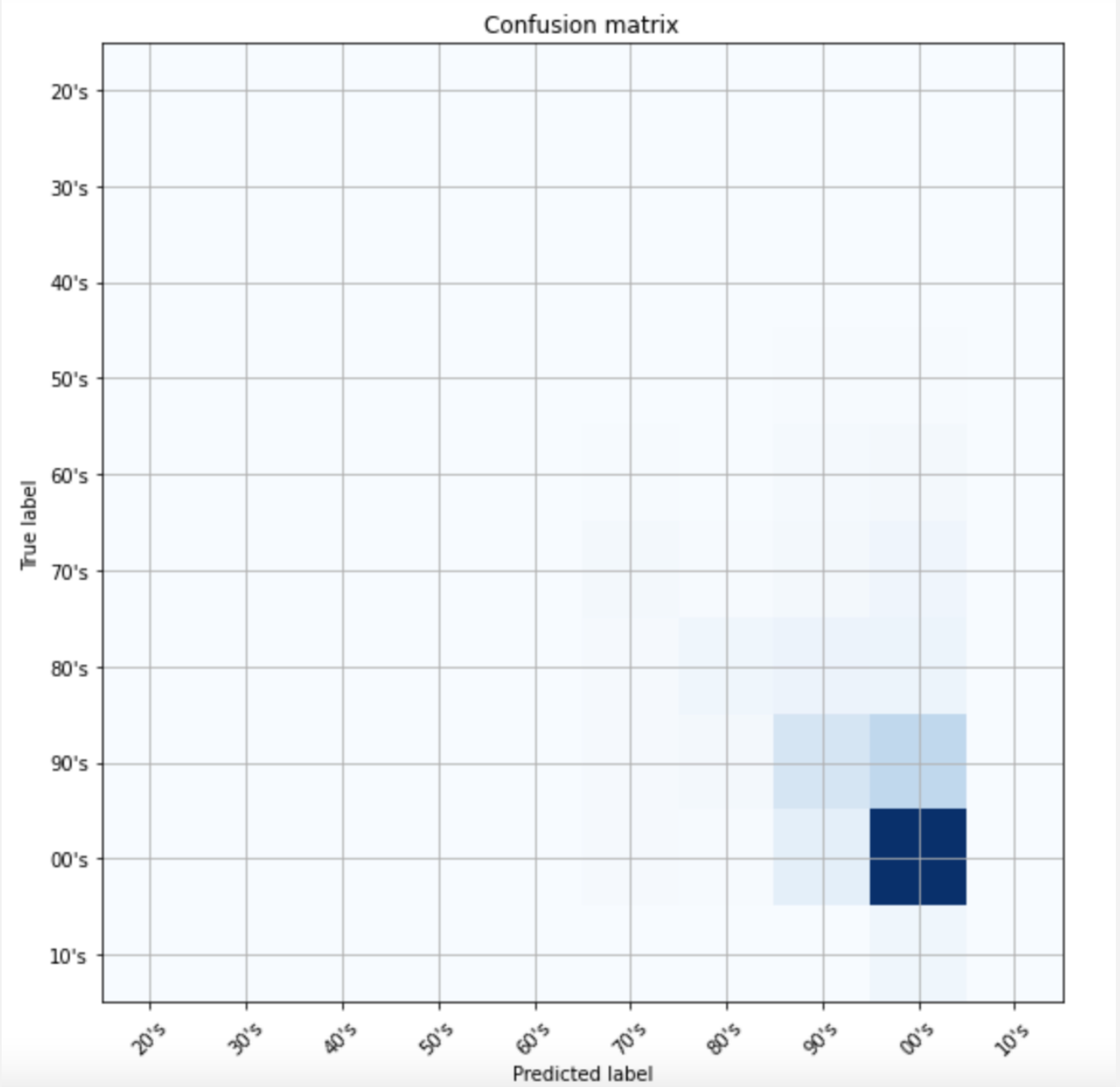




Model - neural network

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 :

<u>The best model is definetely the neural network with the grouped data</u>