# Dataset creation from last.fm or other folksonomy tags

This task was about getting  a relevant subset of a Folksonomy of tags in order to upload a dataset into AcousticBraninz so it can be trained and classified.

In our case, we used the provided Last.fm CSV file in order to to make a “Happy”/”Sad” mood classifier.

## Input: Folksonomy

As stated before, we worked from the given file 'acousticbrainz-2015-01-lastfm-tags.csv' wich contains 823033 instances, and 39 columns wich represent it's MBID followed by pairs of data the first item in a pair is a tag, and the second item is its count in that recording. These counts are normalised per recording, that is, a count of 100 means that it is the most common tag in that recording, not that it has been tagged with this tag 100 times.
As tagging is Last.fm is free, meaning that user can tag any facet of the sound (mood, genre, artist characteristic, random thoughts….) and not moderated it is prone to manipulation by the site's users, most often resulting in some disagreements and random tags.
For this reason we tried to use the simplest possible approach and also trying to consider only the sounds that have maximal counts for the tags of our interest.

M

823033x39 cell

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| 91 | 00101edd-3a4f-47eb-aac1-767584291e0a | rock | 100 | indie rock | 51 | indie | 36 | Canadian | 27 |
| 92 | 0010283d-13db-4af8-873e-447d01fb9dbf | motown | 100 | doo wop | 100 | bateman | 100 | RB | 50 |
| 93 | 00104142-e5fb-43aa-acb3-fd71523355aa | female vocalists | 100 | alternative | 100 | indie | 100 | rainy days | 75 |
| 94 | 001082fc-517d-4c83-a753-d3a4117ff94d | turkish | 100 | eastern martial | 100 | | | | |
| 95 | 0010a64e-d781-4596-b470-93ad48467e26 | | | | | | | | |
| 96 | 0010a9a5-c4a9-4309-9e7c-58a0f7caaa7d | mathcore | 100 | experimental | 53 | Progressive metal | 40 | metal | 36 |
| 97 | 0010bf90-3721-4dd1-8a10-17bfb9bf0705 | SuomiPop | 100 | tyttoenergiaa | 100 | | | | |
| 98 | 0010f597-1d3b-4401-ae4e-68ed147b340c | Classical | 100 | guitar virtuoso | 100 | instrumental | 66 | rock | 33 |
| 99 | 001159ec-c7c1-472b-acfc-3266d40ab017 | new age | 100 | gregorian | 41 | easy listening | 16 | loneliness after dusk | 8 |
| 100 | 001161df-758b-40e0-8252-eb30f6d80978 | | | | | | | | |
| 101 | 0011b0ad-fe30-4409-9a7c-5a091509627a | | | | | | | | |
| 102 | 0011b95f-9f95-4c9a-89d3-fdaefdf0c1b6 | instrumental | 100 | | | | | | |
| 103 | 0011c1fa-fa85-4d03-a45c-c9e70903b1c4 | | | | | | | | |
| 104 | 0012070e-78ce-419d-b7d7-a0f4408d8083 | | | | | | | | |
| 105 | 001215c7-6b28-40a4-be34-ec68714a9412 | instrumental | 100 | Soundtrack | 100 | neo classical | 100 | | |
| 106 | 00122052-87d2-468e-9ae0-a5d8d06644a2 | alternative | 100 | Comets And St... | 50 | magic | 50 | 10 of 10 stars | 25 |
| 107 | 00122eda-1e2a-4dbb-8447-abbb130f5487 | | | | | | | | |
| 108 | 00126780-805d-474f-9209-47fe2059d93b | rock | 100 | Mark Knopfler | 50 | guitar | 44 | singer-songwriter | 33 |
| 109 | 00128419-842d-470e-8bdd-869ef90ba8dc | jazz | 100 | free jazz | 100 | experimental | 100 | Avant-Garde | 10 |

*Figure 1: Fragment of the parsed CSV Folksonomy.*

## Taxonomy

We decided to build a 'mood' dataset, trying to establish a characteristic mood or sensation that most users can agree ton.
As stated before, Last.fm Folksonomy could have a a huge variety of tags and event disagreements and contradictions in the same piece so we decided to keep it simple and establish a very trivial and common taxonomy in order to extract relevant information from our Folksonomy with words most users know and use frequently.

'Happy' and 'Sad' are the two classes we are going to use. We are looking for the presence of this exact tags in the whole database and try to construct the database from here.
We know that emotions can be expressed in many ways; and even our selected emotion can be themselfs expressed in different ways, but we considered them as the standard, common and robust way to express these opposed moods.

| | TEXT | TEXT |
|---|---|---|
| 989 | 07de41bd-1d03-45ca-851c-53c06a76c129 | happy |
| 990 | 81af8579-d141-444e-96ae-6d52e6aec3f2 | happy |
| 991 | e3b05559-46db-49d7-af5a-a8500d3f51ad | happy |
| 992 | d1241699-3656-4f32-9786-bf6b8fad4339 | happy |
| 993 | 85a28335-a4a3-4b93-98ec-9791405eccff | happy |
| 994 | 70a47c4a-c31e-4649-bfc2-0d9ee7b3a9b4 | happy |
| 995 | 66797077-8367-43b7-befd-22f940b69354 | happy |
| 996 | 576acff0-623e-453d-9c5c-2ab6b92215bf | happy |
| 997 | dbb042d2-b719-48c5-a196-3344eebff442 | happy |
| 998 | 3d2d566d-04a6-4ca4-8dad-fe3a1060367a | happy |
| 999 | e4a8bba2-1667-4e59-9cc1-c67c065a8e48 | happy |
| 1000 | b6225eda-afc1-4126-8b05-51df594618aa | happy |
| 1001 | 9f6ecd98-ef20-4b6c-87b6-47e8d7f0bcb6 | sad |
| 1002 | b092274d-2bbf-4e3d-9838-9c0ec4644bc2 | sad |
| 1003 | 3a946ab4-e1cd-4463-915b-01ce64d33418 | sad |
| 1004 | b1f6d99d-88ec-4d78-b9f8-079853c21ab8 | sad |
| 1005 | e0f7c9ab-8b56-42c7-8e2e-d0e77964d04c | sad |
| 1006 | d2168704-a833-4a8a-9645-e392268a0baf | sad |
| 1007 | a508504d-f118-4cc4-ab0f-cb7a2ab1c325 | sad |
| 1008 | 7bb43407-3b7a-4cd9-b78c-6b267ebca96b | sad |
| 1009 | dd13f302-9997-41c0-9c5c-097de1292f51 | sad |
| 1010 | acdda53e-f9a1-468a-8e55-aded23a26f6b | sad |
| 1011 | 9bcb27be-c5ae-4248-a7df-114c79858063 | sad |
| 1012 | c3267370-7e37-4e9c-b9f0-46eca788068c | sad |
| 1013 | d9a01219-0a37-4eb7-af98-fc1eed863f24 | sad |
| 1014 | hace31c4-1e72-4a82-8cb2-5751d87ec06f | sad |

*Figure 2: Fragment of the resulting dataset.*

## Procedure

In order to process the Folksonomy to extract relevant instances for our taxonomy we used MATLAB.
Next, we are going to detail the followed process, together with some key pieces of code:

1. We first load the CSV Folksomy, define our Taxonomy and the number of instances we want to extract for each class:

```
 M = csvimport('../CSV/acousticbrainz-2015-01-lastfm-tags.csv');
%Importing CSV database.
 tax = ['happy';'sad  '] % Taxonomy
 N = 1000; % Number of desired instances for each class.
```

2. Then for each class in the taxonomy ('keyword' in our code), we first perform a search for it in the table:

```
indices= find(strcmpi(keyword,M)); % Search the keyword in all the cells.
j = double(idivide(int32(indices),size(M,1))+1); % Get column list
i  =  mod(indices,size(M,1));   % Get row list
```

3. All instances that contain our word are putted into a reduced version of the database, containing only MBID, the keyword and it's count.
   We noted that some instances appear more than one time in the Folksonomy so in this step we also filter them.

```
while k <= length(i)
   idx = find(strcmpi(M{i(k),1},keywordCells), 1); % Checking for
      duplicated ids
 if (isempty(idx)) % If current id is not present in new table
  keywordCells(n,1) =  M(i(k),1); % 1st column: musicbrainz id
  keywordCells(n,2) =  M(i(k),j(k)); % 2nd column: keyword value
  keywordCells{n,3} =  cellfun(@str2num,M(i(k),j(k)+1)); % 3rd column:
  keyword count
  n=n+1;
 end
 k=k+1;
end
```

4. Then, we are going to keep only the defined number of instances four our final database (N=1000). We are ordering our subset decreasingly by considering the keyword count, and take the first N instances, so we are selecting those were our tag is more relevant or prominent:

```
Y = sortcell(keywordCells,3); % Order rows  by considering keyword count.
fY = flipud(Y); % As sortcell orders increasing, we flip to get it d
ecreasing

picked = fY(1:N,:); % Our subset will be the N first instances in our
ordered dataset.
```

5. Following the specifications we will finally generate a CSV file containing MBIDs for all selected instances together with their belonging class. So in our case it will be a 2000x2 table.

```
 db = cat(1,happyDatabase(:,1:2),sadDatabase(:,1:2)); % All instances in
the same table with their class
cell2csv('../CSV/dataset.csv',db);  % Export table into CSV file
```

## Dataset submission and training

With the final CSV file generated containing 2000 instances (MBID's) together with their associated class (1000 of each) we submitted the dataset into our AcousticBrainz profile.

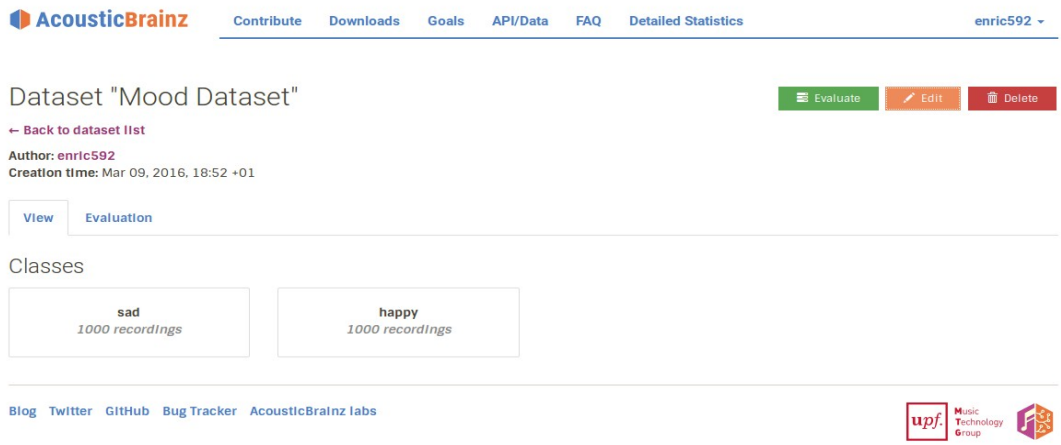


*Figure 3: Screenshot of our AcousticBrainz Dataset.*

We then started the process to automatically train and classify our dataset.

## Results and conclusions

After evaluating in AcousticBrainz we obtained the following results:

- Happy Instances well classified: 86.5%
- Happy Instances classified as sad (error): 13.50%
- Sad instances calssified as happy (error): 16.33%
- Well-classified sad instances: 83.67%
- Overall Accuracy: 85.09%

**Accuracy:** 85.09%

| | Predicted (%) | | | | Actual (%) |
|---|---|---|---|---|---|
| | happy | sad | | Proportion | |
| happy | 86.50 | 13.50 | happy | 50.05 | |
| sad | 16.33 | 83.67 | sad | 49.95 | |

*Figure 4: Table showing our results.*

This results are acceptable considering that the dataset taxonomy was originally defined by random users that can write anything as a tag. Despite this, accuracy is good because we have chosen a very simple and clear taxonomy, and additionally we have taken the first 1000 instances with more consensus on that tag for each class.

Possibly, more could be done in order to consider other ways people could express their mood in relation to a song, but with this simple approach we obtained good results by learning a little more of Folksonomies, Last.fm system and AcousticBrainz.

## References & Links

The developed code is hosted into:
https://github.com/enric592/AMPL-folksonomy-mood

Code is written in Matlab and is located in MATLAB folder. The main script that runs our processes is `dataSelection.m`.
The workspace does not contain the original: acousticbrainz-2015-01-lastfm-tags.csv.bz2 (for space reasons) but it can be found here: http://www.dtic.upf.edu/~aporter/amplab/acousticbrainz-2015-01-lastfm-tags.csv.bz2. It should be placed in the CSV folder in order to successfully run Matlab scripts.

AcousticBrainz generated dataset can be found here: http://acousticbrainz.org/datasets/dc3527cb-aa6d-4689-87dd-f5c2df115bd9

Several Matlab External functions were used:

- `cell2csv.m` : *Sylvain Fiedler, KA, 2004*
- `csvimport.m`: *Ashish Sadanandan, 06 Apr 2009*
- `sortcell.m`: *Jeff Jackson (Ocean Sciences, DFO Canada), Jan. 24, 2007*