Part II.

Implementation

This section focuses on implementation of the previously described models on our own custom task. To do this, we firstly further explore the three previously described datasets to identify necessary cleaning steps, as well as desired tags for each round of testing. We build the necessary data wrangling pipeline and address class imbalance distribution issues and dataset noisiness considerations. A consideration of usable models in the face of many restrictions was conducted, which will also be exposed in this chapter, and the exact design of the preprocessing pipeline, as well as the iteration steps over it, will be presented. Finally, training results and evaluations are presented for all tag wrangling types and all selected models.

6. Data exploration and selection

6.1. General purpose exploration and quality of data

In the previous section we had a summary look at the distribution for the tags of each dataset and the format of these tags to understand a bit better how we can work with them. However, simply checking the format is not sufficient. One must also check for missing data, bad quality data, highly skewed data to set up a cleaning process to have the cleanest data possible. This includes audio features such as length, sample rate, no tags, etc.

This section focuses on the process of cleaning up and selecting relevant data from each dataset before moving on to tag selection for all types of tags.

6.1.1. FMA

The distribution for genres has already been explored for FMA, and the available sub-genres and sub-sub-genres can be found in Annex B. Looking at missing data, the following graph shows missing data proportion per column of the tracks dataframe (Figure 6.1):

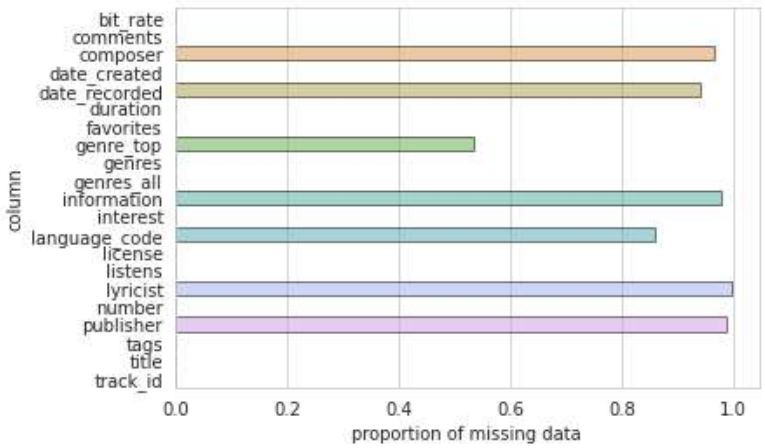


Figure 6.1.: Missing data in the FMA dataset

genre-top, which is the genre most likely to fit the track amongst the previously described top-genres, is missing approx 52% of its data, which means 50% of the dataset has to be discarded if we consider only the top genre. looking at genres and genres-all, which are lists containing the ID of:

- **for genres :** top genres and subgenres
- **for genres-all:** top genres, subgenres and subsubgenres

No data is missing counting null values, but there are empty lists. The same amount of lists are empty for either column, which represents about 2% of the total amount of tracks. Given the size of the dataset, we can

throw these unclassified songs out without too much thought. Furthermore, we have to select songs with genres and subgenres which will be relevant to the model (we will not attempt to classify a subgenre which has only 3 tracks). So, we propose the following processing of tracks regarding genres and subgenres:

- Discard any tracks that do not have at least tag in the genreall columns that is either:
  - Within all the top genres
  - Within the top 10 subgenres
  - Within the top 25 subsubgenres
- For tracks that do have at least one tag within those lists, throw out the tags that are not within those lists.

After genre and subgenre considerations, tags are available for the tracks as well. 22% of tracks have a non-empty tag vector. A first glance at the distribution of track count per tag yields the following distribution (figure 6.2):

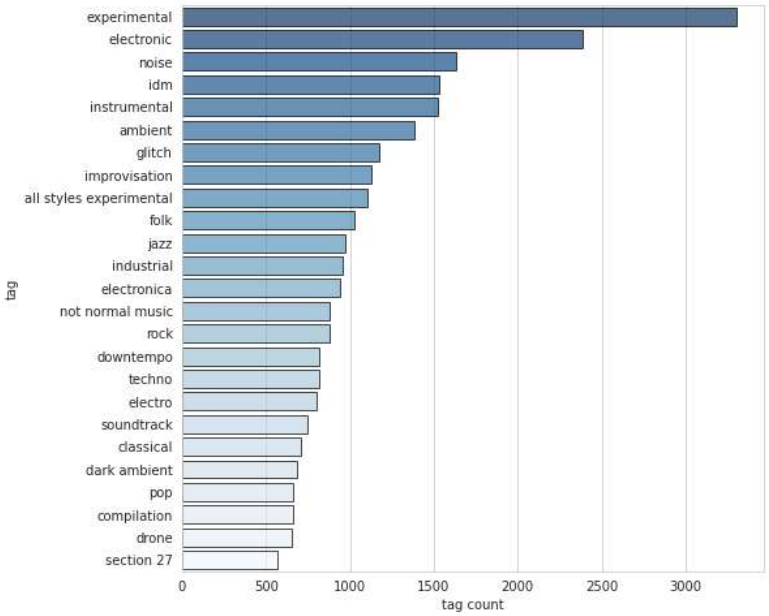


Figure 6.2.: tag distribution for the FMA dataset

Notice that many of the top tags are also genres, subgenres or subsubgenres. Removing these tags non-case-specifically yields the new list of top tags that have not already been considered (6.3):

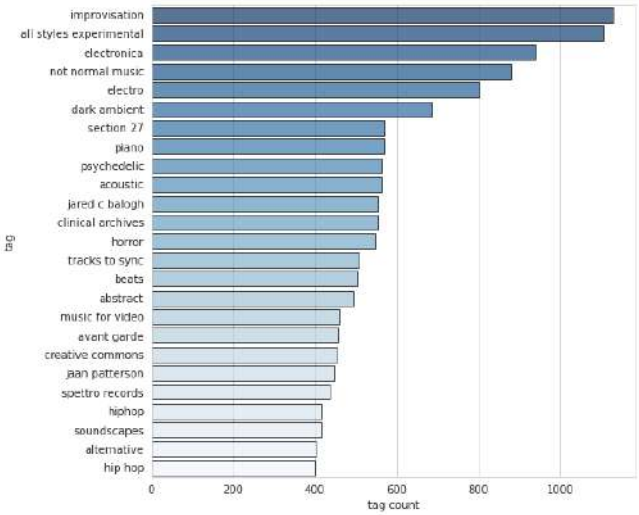


Figure 6.3.: new tag distribution for the FMA dataset

Most of these tags are relevant, and so we also choose to include the taglists of tracks that contain tags from within the top 25 tags to the total label vector of each track. Careful however, as some hand cleaning will have to be done (e.g. hip hop is in the tags list but Hip-hop is in the genres list). So, after all this, we have discarded some tracks and all the tracks that are left have a one-hot encoded label vector of 76 total tags. some stratified subsampling will have to be conducted on popular labels to increase the relevance of unpopular labels with regards to model accuracy. After looking at label generation, the quality of audio as well as outliers regarding audio characteristics is also to be taken into account. In figure 6.4 are shown distributions for:

- track bitrate on all tracks
- Track sample rate
- track duration
- track duration when removing tracks with lengths above 1000 minutes.

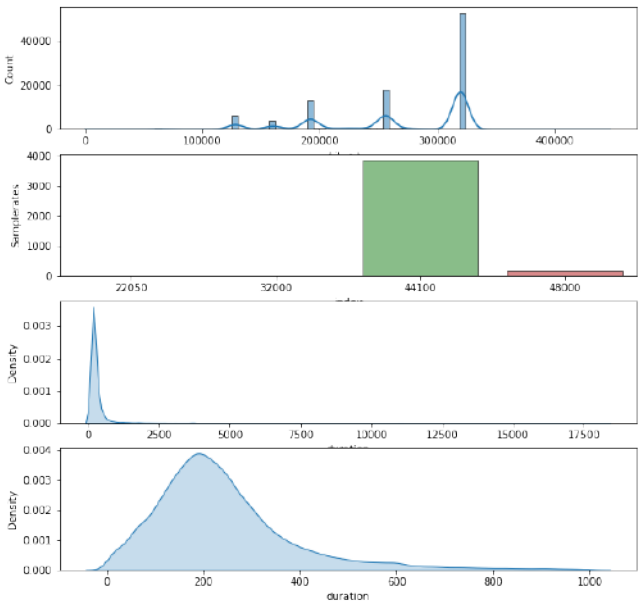


Figure 6.4.: FMA metadata distribution

Most tracks are sampled at 44.1 kHz, which is industry standard. Some other tracks are sampled at 48kHz and 22.5kHz, as well as 32kHz. None of these are far under the sample rates used in the state of the art, so no need to discriminate based on sample rate. Any audio track of duration below 1min and above 10mins (60secs) can be discarded (this represents about 7% of the dataset). Finally, as bitrate has not been shown to influence the performance of audio classification models, we do not filter on bitrate. So, the total selection process for tracks we choose to consider on the FMA dataset is the following :

- Remove tracks of length above 10mins and below 60s
- Downsample tracks with sample rate above 44.1kHz (this is done directly within the preprocessing codebase, no need to do anything in this case)
- Select tracks which only have :
  - genre-all vectors with at least one element within the top-genres, the top 10 sub-genres or the top 25 sub-sub-genres, removing the genre labels that are not within these categories
  - tag vectors that have at least one element within the top 25 tags bar some tags that will be manually cleaned (e.g hip hop or the composer name)

Doing this will yield a clean dataset bar the necessary subsampling which has to be performed before training the model, but this subsampling will be performed later on when all labels from all datasets have been chosen.

6.1.2. MTAT

MTAT has a constant sample rate of 16kHz, which is the lowest sample rate of the three datasets and will thus be our limiting factor for resampling in the preprocessing pipeline. Similarly, all songs are 29.1s long so there is no required filtering on song length. We select the top 100 tags from MTAT to be part of our final dataset. The number of tags per track distribution is shown in the following figure (figure 6.5):

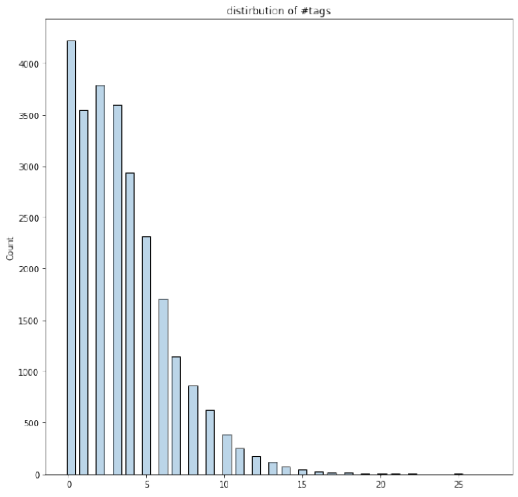


Figure 6.5.: Number of tags per track distribution on the MTAT dataset

Some tracks have more than 15 tags, which are outliers to the rest of the distribution and introduce noise into the dataset, and which we choose to discard. 16% of the available annotated clips do not have any tags, which means we can discard them. We propose the following cleaning pipeline for MTAT: remove songs with 0 tags and more than 15 tags, and discard songs that don't have any tags in the top 100 tags.

6.1.3. MTG-Jamendo

7. Data exploration and selection

For MTG-jamendo, cleaning is rather minimal as well. all songs are the same sample rate (44.1 kHz), so there is no need to oust songs based on sample rate. Since tags are already separated into genres, moods/themes instruments, we select the top 100 genres from the genre section. To check that there are no outliers, the following figure (7.1) shows the dsitribution of tags per track and the duration of the track, which is variable for MTG-Jamendo:

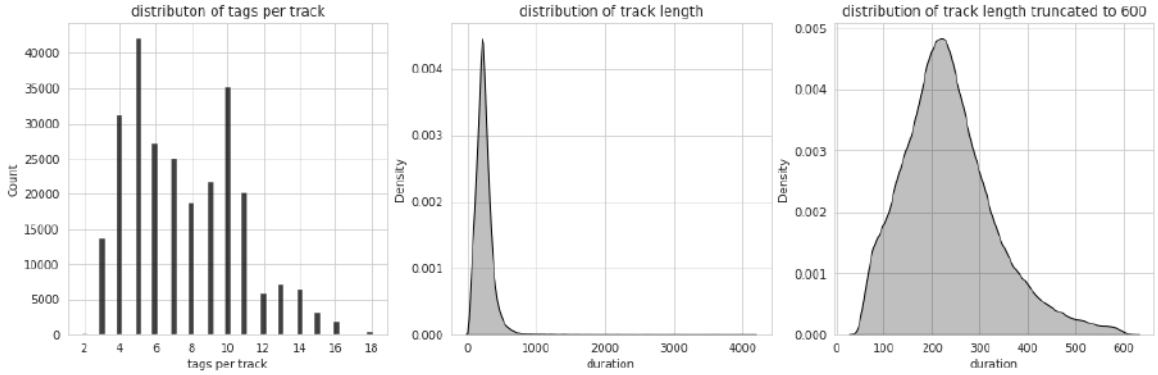


Figure 7.1.: Outlier consideration for the MTG-jamendo dataset

As for MTAT, tracks of length above 600s and below 60s are cut (5% of all tracks). No tracks have 0 tags, which is nice from a data cleaning standpoint.

7.1. Tag selection and distributions before sub/oversampling

We decided at the beginning of the internship to work incrementally with different tag types : Start easy with genres, which are the most mutually exclusive tag set, and allow to get closer to a multiclass single-label task. We then move on to subgenres, then mood/themes/instruments, and finally, we combine all these tags into our own music tagging task relevant to the business use case at groover and different than the typical 50-tag task seen in literature. So, after each round of training and basic fine-tuning for each tag type, we select the tags for the next round of experiments.

Furthermore, tags are noisy in these datasets. for instance, hip-hop and hip hop can bre present within a same dataset. Likewise, male vocals and male singing can be considered as the same tag and are both present in the MTG-Jamendo dataset. This noisiness problem would be very difficult to deal with if we were attempting to classify all tags on all datasets. However, since we select a number of tags to work on at each round of experiments, we peruse the tags from the dataset and do some cleaning by merging tags we consider to be mergeable (for both data augmentation purposes and cleaning purposes). This merging process is also documented in this section.

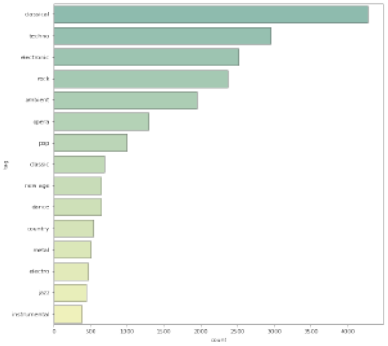
7.1.1. Genres

Among the top tags, the genres are isolated a la mano when not specified - for MTAT for instance. The aim here is to choose an array of genres that are both pertinent to the business case at Groover, and represent a broad array of musical genres to not overspecify the model's objective. the genre distributions for MTAT, MTG Jamendo and FMA are shown below (figures 7.2b, 7.2a, 7.2c)

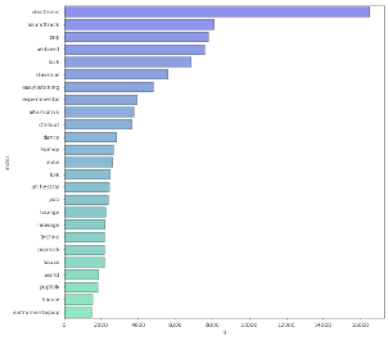
There are many tags here that can potentially refer to very similar things (e.g rock and alternative, hip-hop and rap). Our goal is to merge these tags to encapsulate these concepts better (for genres and for other types of tags as well), but also to reduce noisiness in the datasets (e.g changing 'vocals' to 'vocal' when the 'vocal' tag is already in the dataset tags).

Furthermore, we want to merge the datasets together into one proprietary split of the FMA-MTAT-MTG dataset, which requires some standardization - both in tag taxonomy and format. We use the following processing steps to do this:

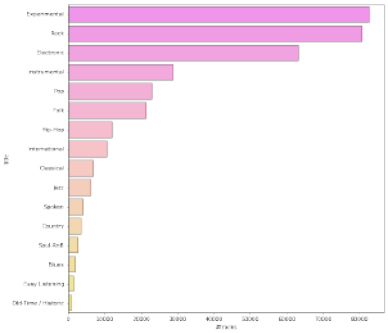
- Select relevant tags among each dataset (overlap is possible here)



(a) top genres distribution for MTAT



(b) Top genres distribution for MTG



(c) Top genres distribution for FMA

- filter each dataset for outliers using the procedures outlined in 6.1
- Build a correspondence dictionnary to reduce tags in each dataset by replacing unnecessary tags with relevant ones.
- standardize the format of the tags (remove special characters, convert to lowercase).
- concatenate the individual datasets.

This will yield a clean dataset with only relevant tags and tracks from all three datasets. For reference, this is conducted in the following on only genres tags for the sake of legibility. 19 total tags are selected and reduced using correspondence dictionaries for each dataset. The wanted tags and correspondence dictionaries for genres for all datasets are shown here. The rest are shown in annex B.

```
"FMA": {
    "wanted_genres": [
        "international","blues", "jazz", "classical", "country", "pop", "rock",
        "easylistening", "soulrnb","electronic", "folk", "spoken", "hiphop",
        "experimental", "instrumental",],
    "genre_merging": {
        "classic": "classical", "dance": "electronic", "electro": "electronic",
        "new age": "ambient", "country": "folk", "opera": "classical",
        "techno": "electronic", "spoken": "hiphop", }, ... }

"MTAT": {
    "wanted_genres": ["techno","classical","electronic","rock","opera","pop",
    "classic","dance","new age","ambient","country","metal","electro","jazz",
    "jazzy", "instrumental","foreign","orchestra","folk",],
    "genre_merging": {
        "classic": "classical","dance": "electronic","electro": "electronic",
        "new age": "ambient","country": "folk","opera": "classical",
        "techno": "electronic","jazzy": "jazz","foreign": "international",
        "orchestra": "orchestral",
    },...}


"MTG": {
    "wanted_genres": [
        "electronic","soundtrack", "pop", "ambient","rock", "classical",
        "easylistening","experimental","alternative","chillout", "dance",  "hiphop",
        "indie","folk","orchestral", "jazz","lounge", "newage","techno","poprock","house",
        "world", "popfolk", "trance","instrumentalpop",
```

```
        "metal","funk", "blues", "rap", "symphonic","darkambient",
    ],
    "genre_merging": {
        "alternative": "rock", "dance": "electronic",
        "lounge": "jazz", "newage": "ambient",
        "techno": "electronic","poprock": "rock",
        "house": "electronic", "popfolk": "folk",
        "trance": "electronic", "instrumentalpop": "pop",
        "world": "international","funk": "soulrnb",
        "blues": "rock", "rap": "hiphop",
        "symphonic": "orchestral","darkambient": "ambient",
    },...}
```

This yields the following new genre distribution with new genres and old genres specified (figure 7.3):

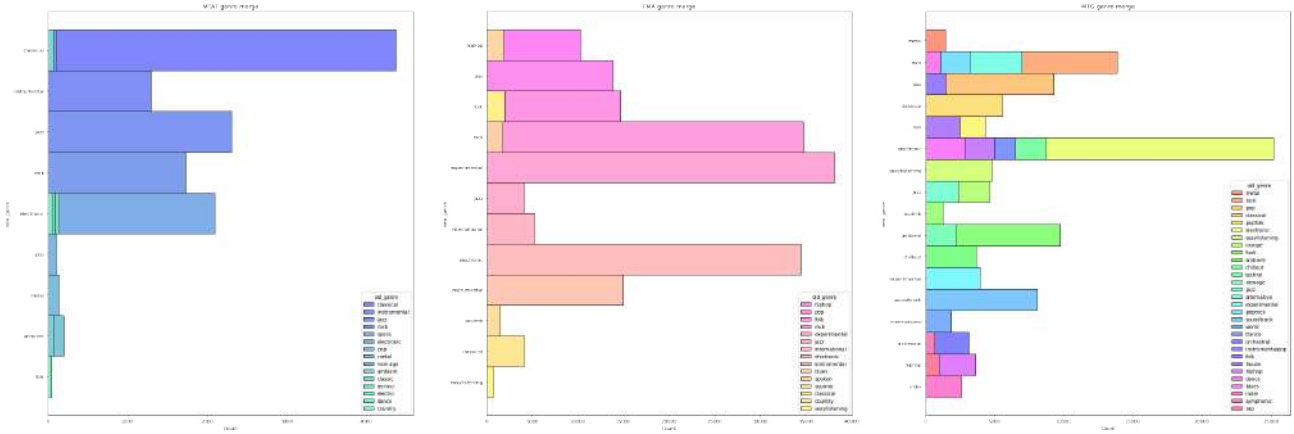


Figure 7.3.: Merging genres in all three datasets

And the genre distribution after dataset merger is shown in following figure (fig. 7.4) Where we can see that despite our best efforts there is still large skewness in favor of some genres, which will require sub/oversampling in future processing considerations

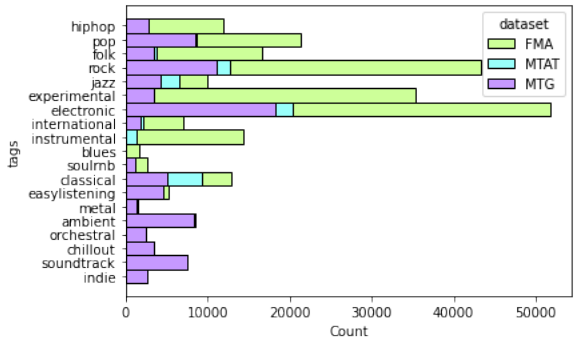


Figure 7.4.: Merging datasets to get final genre distribution

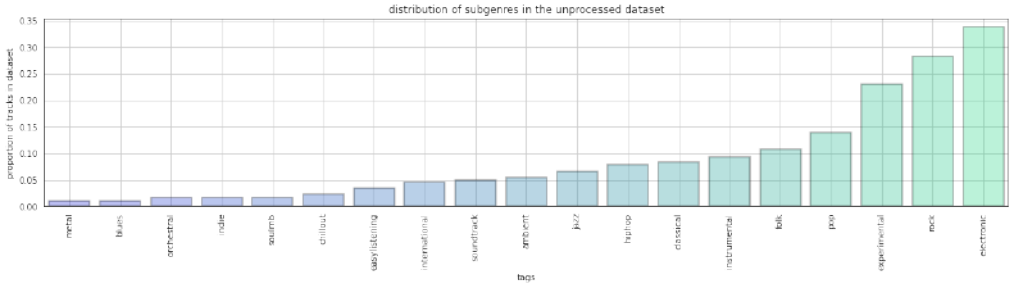
7.1.2. Distribution before resampling for all tags

The same process is repeated for all tag types. At each iteration, relevant tags are taken into careful consideration and the appropriate merging dictionaries are constructed with business appropriacy in mind. A total of:

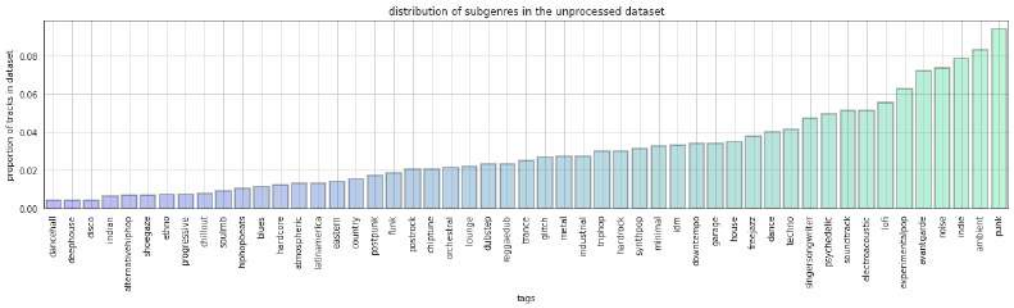
- 19 tags are selected for genres
- 51 tags are selected for subgenres

- 38 tags are selected for moods/themes/instruments
- 80 tags are selected for alltags

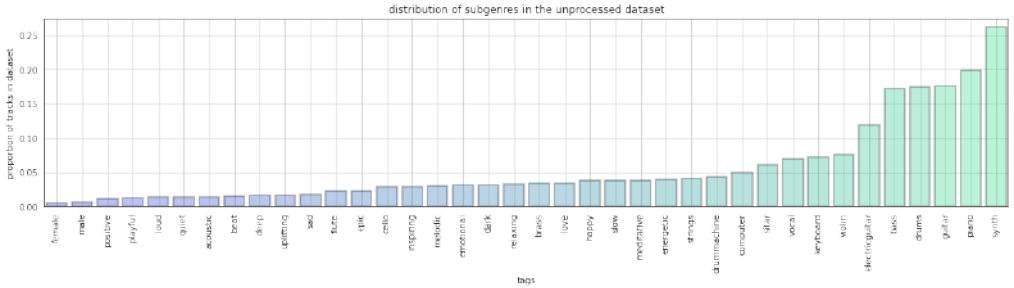
Selection for all tags is slighly different than other selections since we cannot afford to simply add all the previous tags, which would amount to to large of a solution space dimension (113 total tags). We reduce the amount of tags selected as much as possible and end up with 80 tags. Again, all of the wanted tags per type of tag wrangling and per dataset is shown in annex B, as well as the merging dictionaries. The following figures show the final tag distribution before sub/oversampling for each tag type after selection of the aforementioned tags (7.5a, 7.5b, 7.5c, 7.5d). As can be seen in the following figures, all tag types require some form of sub/oversampling to even out the tag distribution, which will be discussed in the following paragraph.



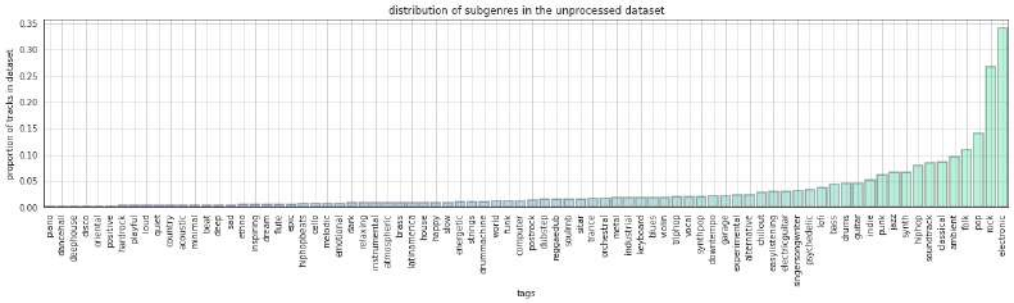
(a) Distribution of genres in merged dataset before resampling



(b) Distribution of subgenres in merged dataset before resampling



(c) Distribution of mood/themes/instruments in merged dataset before resampling



(d) Distribution of all tags in merged dataset before resampling

7.1.3. The need for oversampling and undersampling

The previous section shows that the data is very skewed for some types of tags. For instance, for all tags, "electronic" represents 35% of the total tag volume, which makes the tagging problem for all tags a highly unbalanced problem from a sampling perspective. Notice that in this case, a model which learns to predict electronic 100% of the time would be 35% accurate on a task with 80 tags. This favors model overfitting to the overrepresented tags in the dataset, which is a classic problem in machine learning.

To deal with this, some canonical options exist :

- subsample the majoritary classes before training : this reduces the number of samples associated to the overrepresented classes
- oversample the minority classes : by duplicating some samples of the majority classes, we artifically augment their presence in the dataset.
- attribute higher importance in training to minoritary samples. This skews the model itself towards these underrepresented classes by attributing a higher reward score to them when learning
- creating synthetic data either by performing data augmentation or data interpolation in a representation space for the underrepresented classes.

Our problem requires finesse when adressing this sub/oversampling problem, as it is a multilabel multiclass classification problem. We choose to conduct both subsampling of major classes and oversamping of minor classes. The goal is to flatten out the difference between the count of tags to a given number. for instance, for genres, the minimum number of tracks for a genre is for metal (1500 tracks). The distribution with no sub and oversampling is shown below in figure (7.4) Again, the dataset is highly skewed. To conduct resampling, we still want to conserve original dataset proportions in the new resampled dataset as well as label proportions with comparison to these original datasets. In other words, if metal tags are comprised of 25% MTAT tracks before resampling, we want them to still conform to that metric after resampling). To this end, we conduct stratified sampling based on exploded tags to reduce each tag to 1500 samples (same as the metal tag) and then take the songs from the unsampled dataset which unique identifiers are in the new resampled dataset. This leads to a new distribution shown in figure 7.6

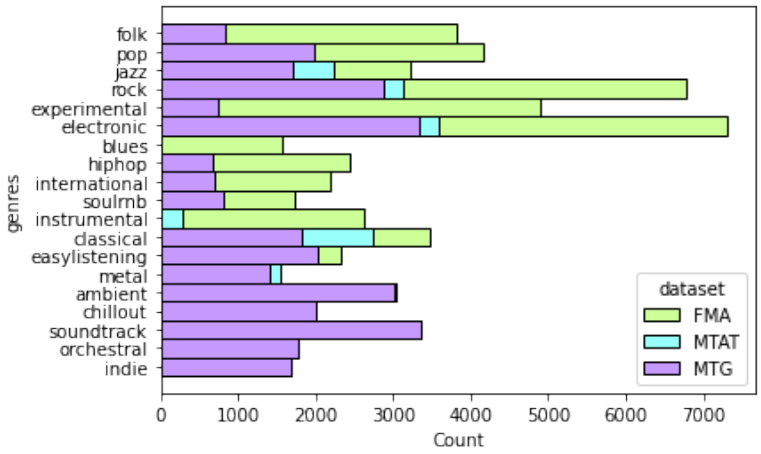


Figure 7.6.: Genre dataset before after resampling

This is also done for the subgenre dataset, the moodthemeinstrument dataset and the alltags dataset. Notice that though the skewness of the dataset has largely diminished **with the new dataset having a max-min ratio of about 6 compared to the previous 33**, the skew has not disappeared altogether. This is due to the multilabl nature of the problem. It would be too artifical of a manipulation to resample only tags with a single label from each genre, and so tracks which are both labeled as metal and rock inevitably contribute to a skewness in favour of rock.