DIT862 Statistical methods for Data Science
CSE, Gothenburg University | Chalmers, Period 2, 2018
Assignment 5: student-defined topics
Jens Madsen, Gleb Bychkov

Attempts of clustering and forecasting music genres using lyrics.

Introduction.

Many companies are more and more interested in text analysis methods. These can be customer requests, internal correspondence, customer chats, internal knowledge base, press review, or social network parsing data.

All this raises a big question, how can you automatically analyze large amounts of data and later use the data and results as a base for decision making procedures.

In order to train these skills and being music lovers we decided to use the dataset of lyrics that is provided by kaggle[1] in an attempts of usage statistical methods, that we learned in this class to cluster and forecast music genres using lyrics of the songs.

Data.

The dataset consists of more than 380 000 observations, each observation shows name of the song, release year, artist, genre and lyrics of the song themself. We are mostly interested in 'lyrics' and 'genre' information. The original dataset presented the following genres: ['Pop', 'Hip-Hop', 'Not Available', 'Other', 'Rock', 'Metal', 'Country', 'Jazz', 'Electronic', 'Folk', 'R&B', 'Indie']. We decided to exclude 'Not Available' and 'Other' categories from the sample as we think there is no special semantic or featured vocabulary, which might statistical methods could analyse.

We also excluded all observations from the sample, that included 'weird' years like 67, 112, 702 and 2038 as we think lyrics written those years might have poor sense.

The next step of cruel cleaning of the data was getting rid of the songs, that do not have any lyrics, as this may violate the goal of the exercise. At this point we are left with 237 421 songs in the dataset. Later, we deleted nearly 30 000 songs more, as they are appeared to be not in English language, the reason for that we will describe further in the report.

The next step in the data processing for the analysis was cleaning the lyrics themselves, we removed punctuation marks, indents (as lyrics is usually written as poems) and stemmed the lyrics using NLTK python package.

We also perform very simple exploratory analysis of the distribution of the genres in the dataset.

[1] https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics

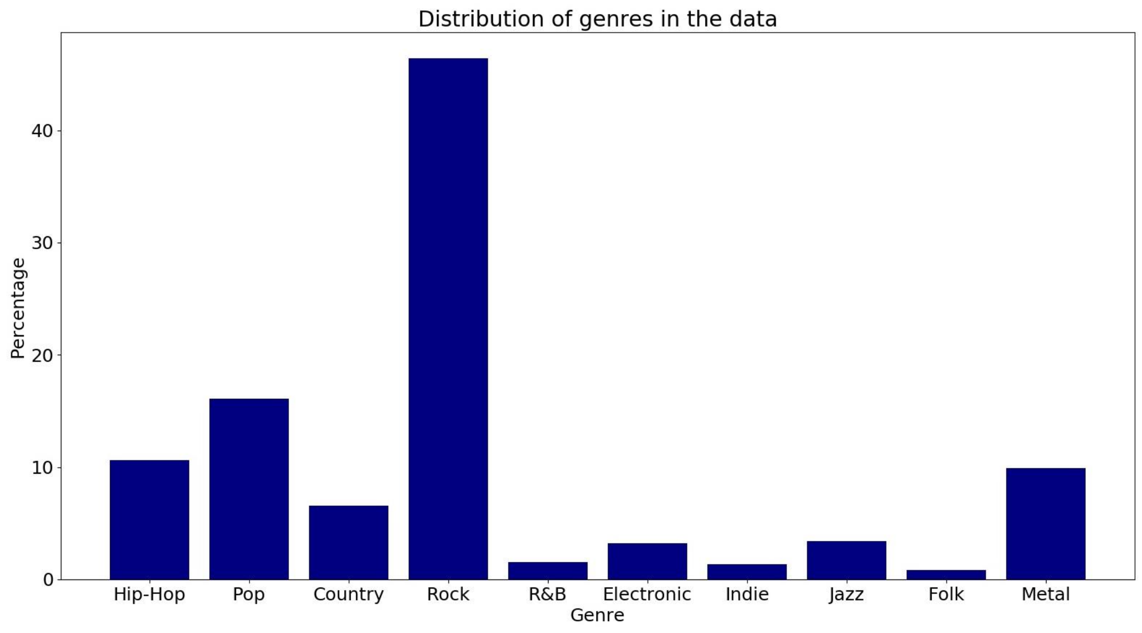


Fig.1. Distribution of genres in the dataset.

Around 45% of all lyrics that are represented as 'Rock' music, the second largest group is 'Pop' music with 17% of the sample. The smallest group in the sample is 'Folk' music, it represents around 1% of the sample.

The distribution is far from the uniform one, which is making us expect some anomalies and discrepancies in the analysis.

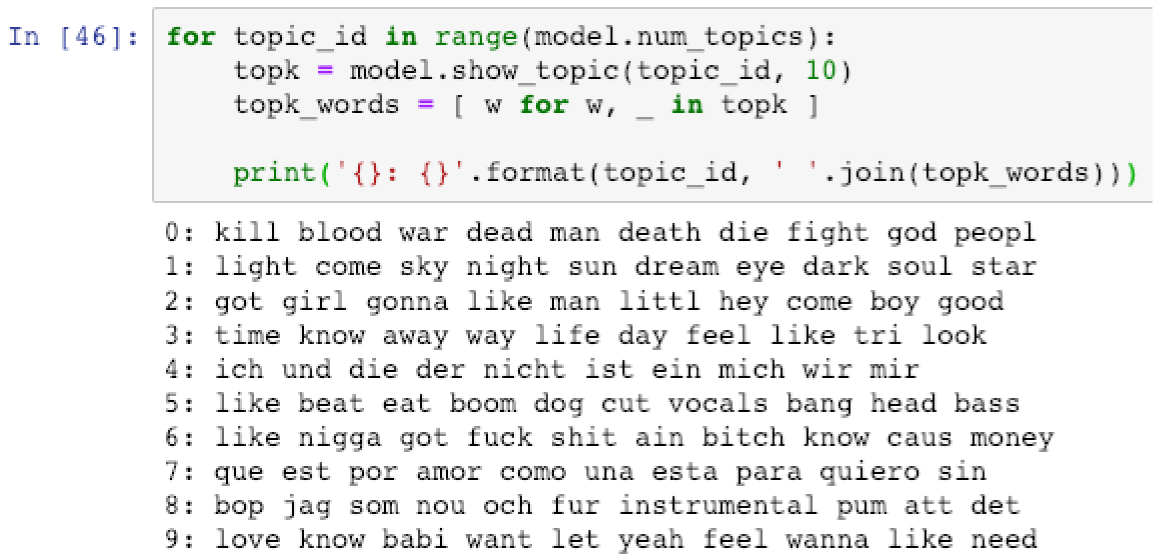
Latent Dirichlet Allocation.
As a first step in the analysis we decided to implement Latent Dirichlet Allocation (LDA) for this dataset. LDA was first presented as a graphical model for topic discovery by David Blei, Andrew Ng and Michael I. Jordan in 2003[2].

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

The code for computing LDA is presented in the attached file 'Text clustering (LDA)'.

The first output that we achieved:

[2] David Blei, Andrew Ng and Michael I. Jordan Latent Dirichlet Allocation Journal of Machine Learning Research 3 (2003) 993-1022, http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf



```
In [46]: for topic_id in range(model.num_topics):
             topk = model.show_topic(topic_id, 10)
             topk_words = [ w for w, _ in topk ]

             print('{}: {}'.format(topic_id, ' '.join(topk_words)))
```

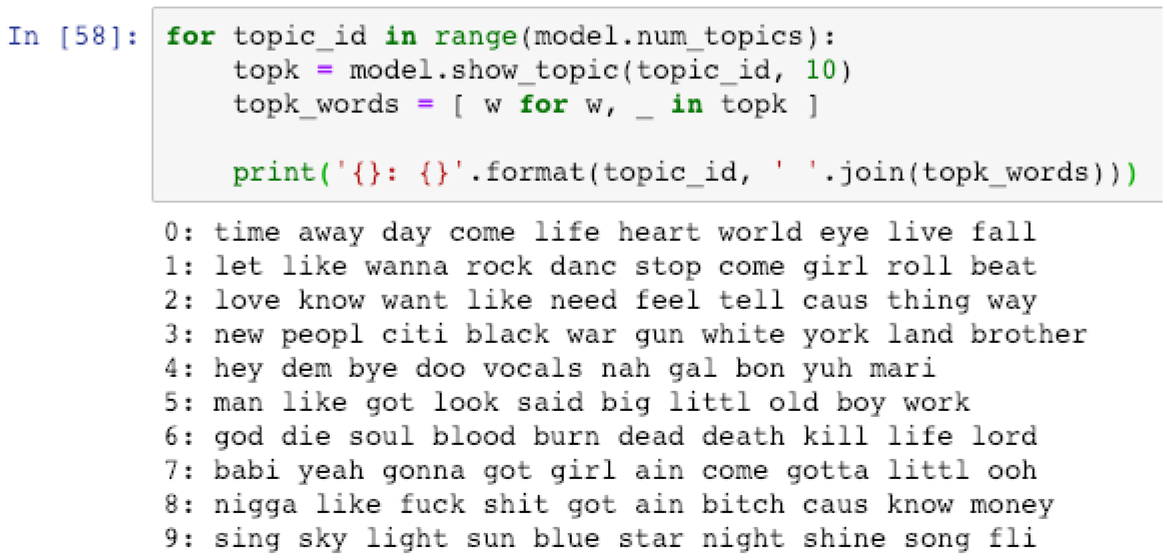
```
0: kill blood war dead man death die fight god peopl
1: light come sky night sun dream eye dark soul star
2: got girl gonna like man littl hey come boy good
3: time know away way life day feel like tri look
4: ich und die der nicht ist ein mich wir mir
5: like beat eat boom dog cut vocals bang head bass
6: like nigga got fuck shit ain bitch know caus money
7: que est por amor como una esta para quiero sin
8: bop jag som nou och fur instrumental pum att det
9: love know babi want let yeah feel wanna like need
```

The first thing that we think is really interesting in this output, that you can actually able to guess the genre in the output. For example: row 0 is apparently 'Metal', row 1 could be 'Folk', and we could also find some other genres in the output. Interesting thing in terms of the results, that was provided by the method, that it actually identified languages as a separate topics in the data (we can try to call them 'German' and 'Spanish' genres in out setup). This is how we identified that the dataset has lyrics in languages, other than English. So, we had to return to the cleaning part again to get rid of the 30 000 songs more.

In order to identify the language, we used langdetect python's package.

As we cleaned the data, we updated the results of the LDA.

The new result:



```
In [58]: for topic_id in range(model.num_topics):
             topk = model.show_topic(topic_id, 10)
             topk_words = [ w for w, _ in topk ]

             print('{}: {}'.format(topic_id, ' '.join(topk_words)))
```

```
0: time away day come life heart world eye live fall
1: let like wanna rock danc stop come girl roll beat
2: love know want like need feel tell caus thing way
3: new peopl citi black war gun white york land brother
4: hey dem bye doo vocals nah gal bon yuh mari
5: man like got look said big littl old boy work
6: god die soul blood burn dead death kill life lord
7: babi yeah gonna got girl ain come gotta littl ooh
8: nigga like fuck shit got ain bitch caus know money
9: sing sky light sun blue star night shine song fli
```

We think that genres could be identified as follows:
0: Indie, 1: Rock, 2: R&B, 3: Jazz, 4: Electronic, 5: Country, 6: Metal, 7: Pop, 8: Hip-Hop, 9: Folk.

The results actually surprised in a way that LDA is giving very intuitive result. Though as some genres like Hip-Hop and R&B, Folk and Indie may be hard to identify as they might use similar vocabulary in their ‘Topics’.

Naive Bayes classifier and forecasting.
Next step in our analysis of lyrics is an attempt of application Naive Bayes (NB) classifier for the analysis and forecast of the genres in the dataset.

We divided our dataset on 2 dataset: training dataset, that has 80% of observations and the rest of the dataset was used for the evaluation of the method.

The accuracy that we got using NB is roughly 26%, which we think is somehow low. In order to analyse the reasons of such low accuracy, we computed ‘accuracy matrix’, at least we are calling it like that.

The results of the NB classifier:

	Hip-Hop	Pop	Country	Rock	R&B	Electron	Indie	Jazz	Folk	Metal
Hip-Hop	3030	201	3	262	10	71	1	27	5	90
Pop	187	951	106	1013	37	179	23	51	5	60
Country	40	502	1668	2012	35	68	42	187	18	28
Rock	40	287	46	1846	31	62	39	23	9	159
R&B	622	1230	223	2395	108	123	40	164	10	87
Electro	229	353	12	701	9	169	18	23	9	66
Indie	704	1756	311	6978	118	487	442	196	72	510
Jazz	53	752	251	1047	26	72	10	643	8	19
Folk	186	522	211	2210	12	84	98	146	122	309
Metal	107	257	27	1946	19	98	30	18	12	2409

The rows of the matrix represent guesses of the classifier, while columns ‘correct’ genres. In that case, main diagonal of the matrix can represent correct guesses, while elements outside main diagonal incorrect ones. By looking at any element outside the main diagonal, we can also say if which genre was labeled by the classifier incorrectly. For example, looking at second row and first column of the matrix gives us number 187, which means that classifier 187 times labeled text as Pop, though true genre was ‘Hip-Hop’.

We also decided to analyze the output, one of the reasons of bad performance could be different distribution of the genres in the sample: if the data was sorted by the genres (all observations of one genre in the beginning, another only in the end of the sample), then the results could be distorted by that issue.

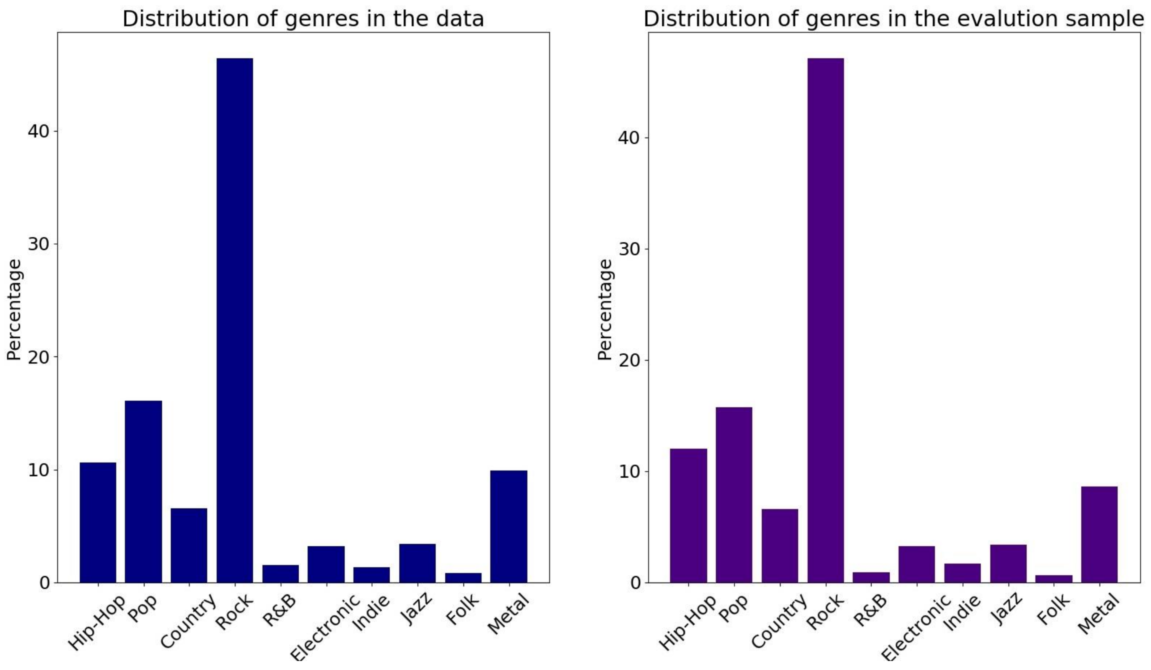


Fig.2. Distributions in test and evaluated samples.

As can be seen from the picture, even without randomizing the data, distributions in train and test samples are the same.

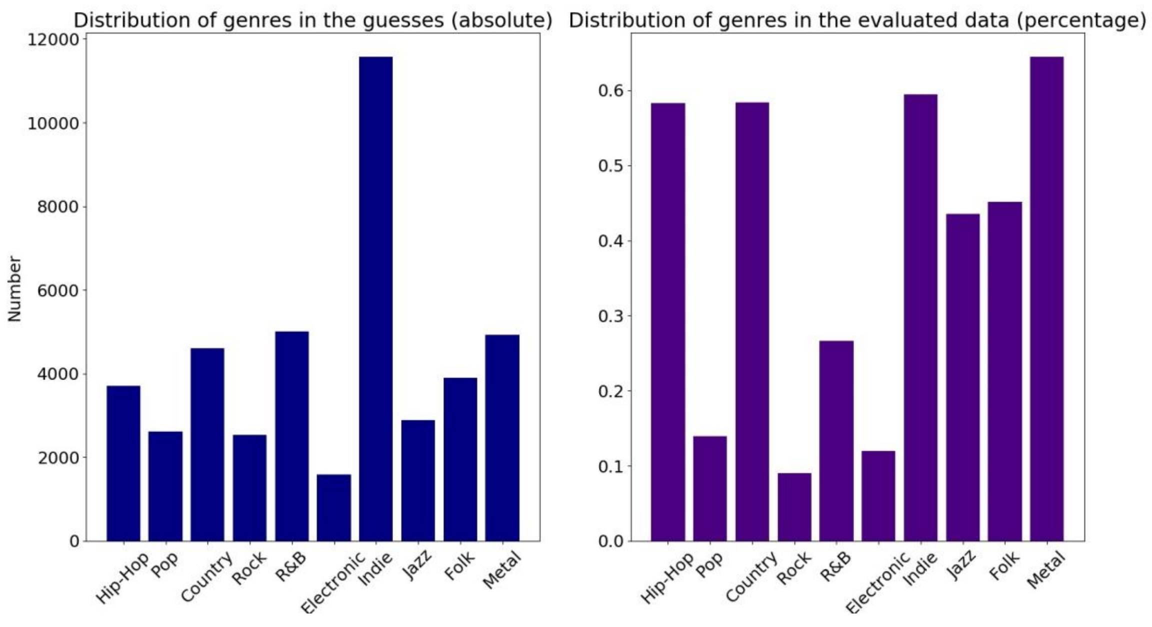


Fig. 3. Distribution of guesses in absolute and relative terms.

The distribution of guesses if also does not seem to be uniform. Model tries to label 'Indie' genre much more often than any other genre, the most 'unlabeled' genre in the guesses is 'Electronic music'. If we would move to the percentages of guesses, we can see, that 'Metal' genre is guessed better than any other genre, also good percentages in 'Indie', 'Country' and 'Hip-Hop', their guesses around 55-60%. The accuracy is highly lowered by very low percentage of 'Rock' guesses. As this genre has 45% representation in the sample, this is exactly the genre that 'kills' the accuracy for NB.

We think that happens, because probabilities of words occurring in the model for 'Rock' categorie are low, compared to other genres. First of all, it has the highest proportion in the data representation and also, this genre is really diverse in terms of vocabulary. 'Small' genres have smaller samples, which lead us to higher probabilities of words occurring in them. Though 'Folk' has the smallest sample size, it might have very specific vocabulary used in it, as 'Indie' genre is the second smallest sample, though it might be the case, that words, that are used in it are much more general and also often used in other genres.

Support Vector Machine classifications.

Lastly in our analysis we wanted to try different models to see if we could improve our results in forecasting genre based on the lyrics. This was done because we were unhappy with the results of both the Naive Bayes model and the LDA model.

To start things off we used attempted to classify the lyrics using a Linear SVC, also known as a Linear Support Vector Machine. SVC's are supervised learning models, which means that the data should be labeled whilst being trained. We got the best results using an n gram range of one, meaning that the model only recognizes the frequency of words during training and therefore does not care what words are before or after in order.

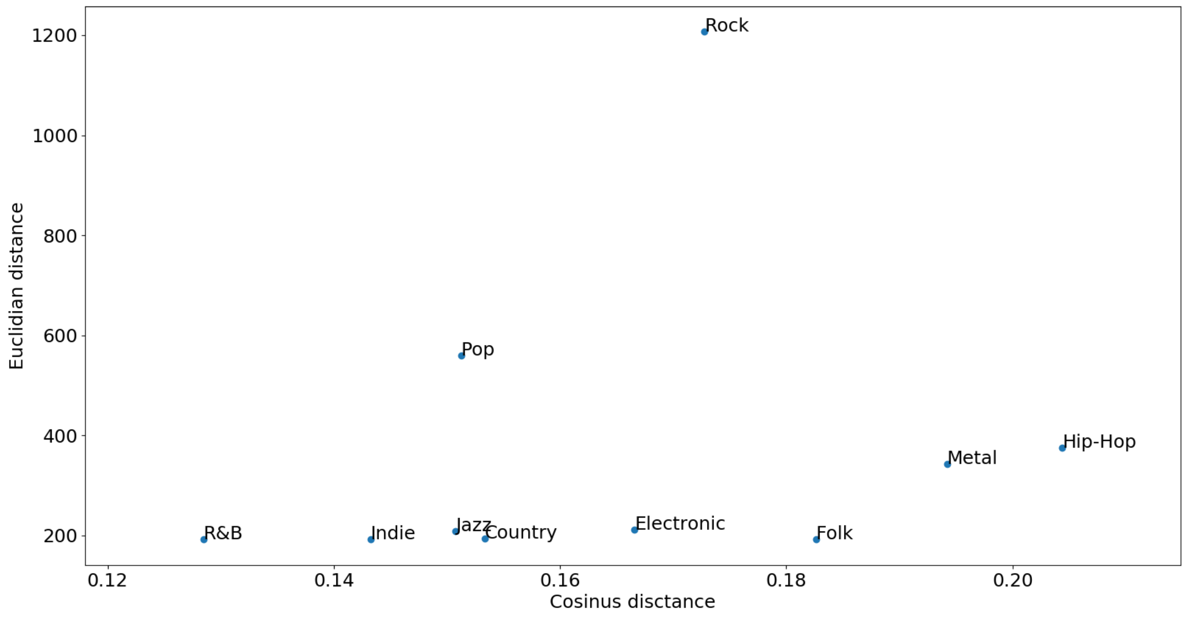
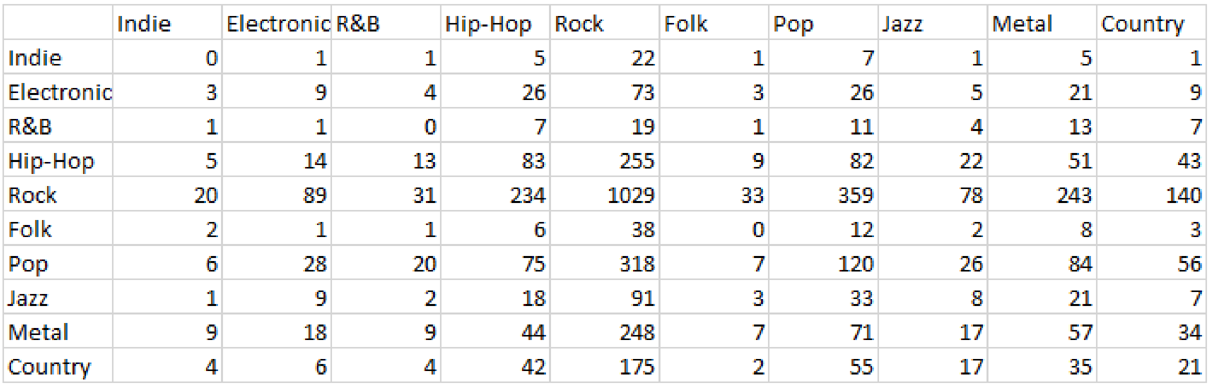


Fig.4. Distance between genres.

After vectorizing the lyrics, we made 10 vectors by summing up vector per lyrics. We computed Euclidean distance and cosine distance between vectors that represent genres and vector of ones (artificial genre that uses all words in a sample only ones). Y-axis is Euclidian distance between genres, it actually somehow replicates distribution of the genres, so if distribution would be somehow uniformed, visually we can only interpret as the 'size of the data'. X-axis shows cosine distance. We can have a hypothesis, that 'Indie', 'Pop', 'Jazz' and 'Country'. 'Metal', 'Folk' are close too. The surprising thing that 'Hip-Hop' and 'R&B' are the most distant genres.

Implications
During the trials of different models we had problems with memory. We couldn't run our notebooks because they would shut down due to our computers not having enough memory to compute the forecasts. Therefore, we downsized the dataset from ~220 000 rows to 25 000 rows. This can of course have implications on the results and also makes it hard to judge whether one model or another works better.

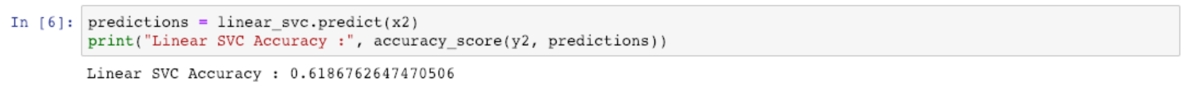
Results



	Indie	Electronic	R&B	Hip-Hop	Rock	Folk	Pop	Jazz	Metal	Country
Indie	0	1	1	5	22	1	7	1	5	1
Electronic	3	9	4	26	73	3	26	5	21	9
R&B	1	1	0	7	19	1	11	4	13	7
Hip-Hop	5	14	13	83	255	9	82	22	51	43
Rock	20	89	31	234	1029	33	359	78	243	140
Folk	2	1	1	6	38	0	12	2	8	3
Pop	6	28	20	75	318	7	120	26	84	56
Jazz	1	9	2	18	91	3	33	8	21	7
Metal	9	18	9	44	248	7	71	17	57	34
Country	4	6	4	42	175	2	55	17	35	21

Once again, the rows of the matrix represent guesses of the classifier, while columns 'correct' genres. These results were obtained using a linear SVC model. As the matrix shows, *Rock* is a common guess regardless of genre. We think that it is because such a big amount of our data is represented by Rock music and therefore more words are associated to rock music inside the model.

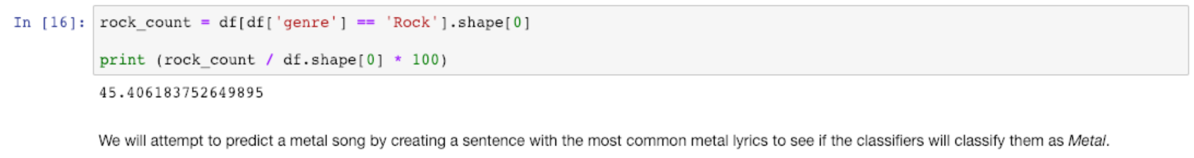
The accuracy of the model is represented in the picture below.



```
In [6]: predictions = linear_svc.predict(x2)
        print("Linear SVC Accuracy :", accuracy_score(y2, predictions))

        Linear SVC Accuracy : 0.6186762647470506
```

As shown above the model has ~62% accuracy. Of all the lyrics Rock music stands for ~45% of the data sample as shown in the image below.



```
In [16]: rock_count = df[df['genre'] == 'Rock'].shape[0]

         print (rock_count / df.shape[0] * 100)

         45.406183752649895
```

We will attempt to predict a metal song by creating a sentence with the most common metal lyrics to see if the classifiers will classify them as *Metal*.

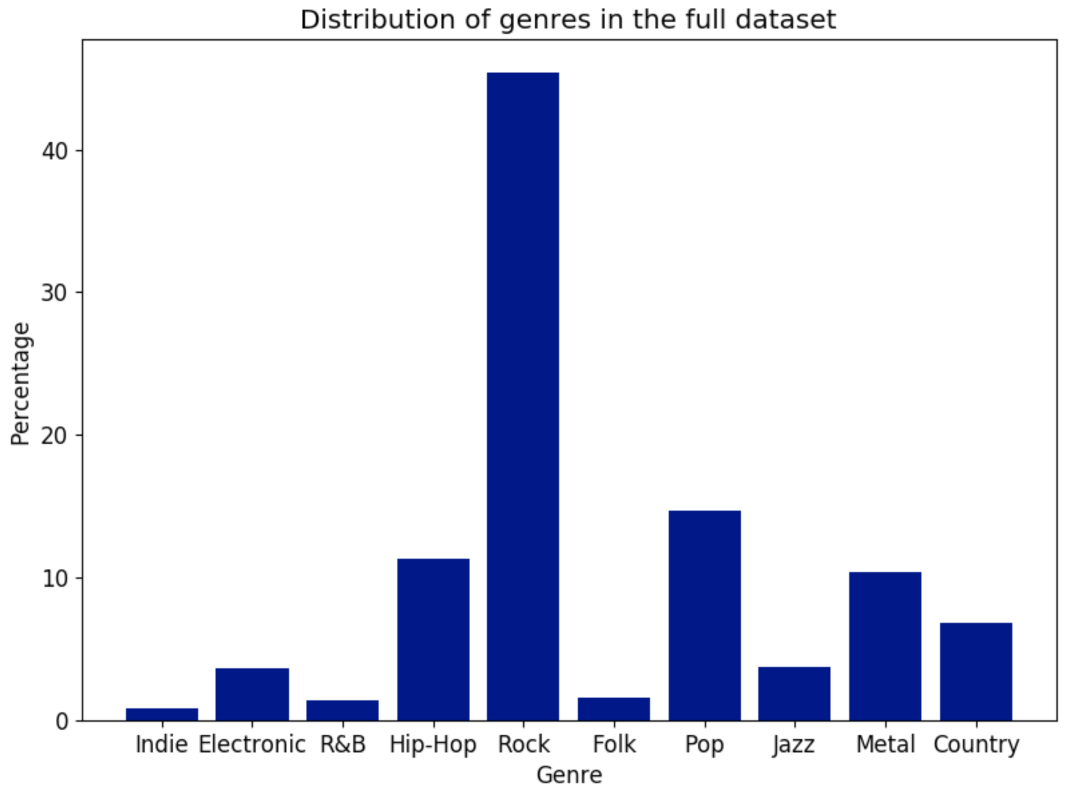


Fig. 5. Distributions of genres in the full dataset.

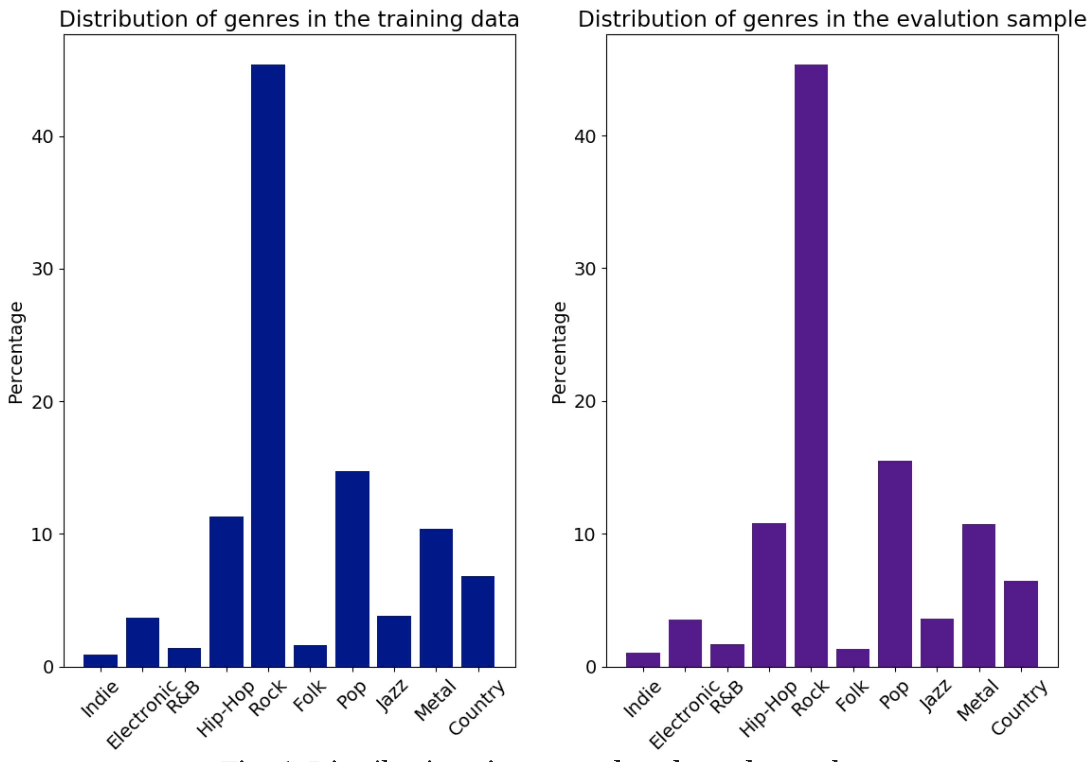


Fig. 6. Distributions in test and evaluated samples.

Even though we are using SVC in a subsample of the dataset, due to memory capabilities of the computers, distribution in subsample replicates the distribution of the whole sample.

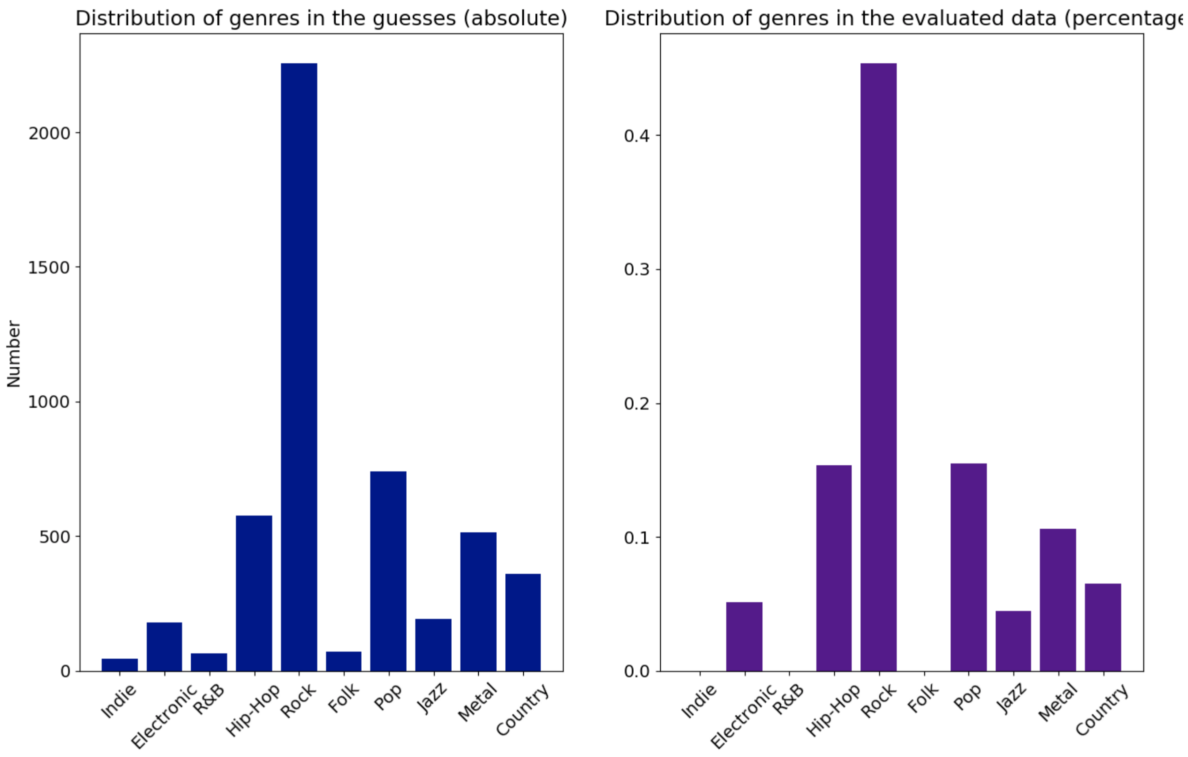


Fig. 7. Distribution of guesses in absolute and relative terms.

It seems as if the Linear SVC model replicates the distribution of the data in its guesses, as we were discussing in the NB part of the assignment. NB has a lower accuracy because it poorly guesses the 'Rock' genre. In this classifier the exact opposite is happening, classifier overestimates guesses of 'Rock' music, which in return returns higher overall accuracy.

Indie music in NB classifier had accuracy around 60%, here it is 0% as none of 'Indie' song was labelled correctly. It seems that one the strengths with this classifier is that it has a higher overall accuracy compared to Naive Bayes model. One of the possible weaknesses with the Linear SVC model is that some genres were never guessed at all, meaning that it has a low accuracy when predicting these labels (genres in the case of this experiment).

Conclusions

The difference between the Naive Bayes model and the Linear SVC is that the Linear SVC uses the inverse to calculate the probabilities. If you want to use a model that is good at guessing underrepresented labels, Naive Bayes is probably the model to use. If you want to guess the label that is the most represented in a dataset the Linear SVC is probably the model that you want to use.

The problem in classification of genres using lyrics of songs is the subjectivity and ambiguity of the categorization used for training and validation. Often genres don't even correspond to the lyrics of the music but to the time and place where the music came up or the culture of the musicians creating it.

Further discoveries (questions to ask)

As we feel problem with classifications that we faced working on this assignment may be connected to non-uniform distribution of genres in the sample. We would like to discover a method that may overcome this problem.

As we faced the problem of lacking memory in order to use all the sample that we have in the set, we are really looking forward Techniques for Large-scale Data course of our program.

We also would like to try some combination of Markov-chain and LDA methods in an attempt of synthesizing lyrics for the genre. We also keeping in mind that maybe author could add some information for the forecast of the genre.

As we look at different kernels on kaggle website, it might be useful to try to use some machine learning technique in the analysis of the dataset.

References

1. David Blei, Andrew Ng and Michael I. Jordan Latent Dirichlet Allocation Journal of Machine Learning Research 3 (2003) 993-1022, http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf
2. https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics
3. Visualizing music: The problems with genre classification, https://mastersofmedia.hum.uva.nl/blog/2011/04/26/visualising-music-the-problems-with-genre-classification/?fbclid=IwAR3h54tDGHFNf4YGOybG33Gy_Kh0faAKyjwr--IPaYgxmNmk18O89520QrU